

UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA

Veljko Vujčić

**ORKESTRACIJA ALATA I
HETEROGENIH TOKOVA PODATAKA
U ASTRONOMIJI VREMENSKOG
DOMENA**

Doktorska disertacija

Beograd, septembar 2026

UNIVERSITY OF BELGRADE
FACULTY OF ORGANIZATIONAL SCIENCES

Veljko Vujčić

**SOFTWARE ORCHESTRATION OF
HETEROGENEOUS DATA STREAMS
AND PIPELINES IN TIME-DOMAIN
ASTRONOMY**

Doctoral Dissertation

Belgrade, September 2026

Mentor:

dr Slađan Babarogić, mentor, redovni profesor
Fakultet organizacionih nauka, Univerzitet u Beogradu

Članovi komisije:

dr Gordana Savić, redovni profesor
Fakultet organizacionih nauka, Univerzitet u Beogradu

dr Nenad Aničić, redovni profesor
Fakultet organizacionih nauka, Univerzitet u Beogradu

dr Željko Ivezić, redovni profesor
Department of Astronomy, University of Washington

Datum odbrane:

ORKESTRACIJA ALATA I HETEROGENIH TOKOVA PODATAKA U ASTRONOMIJI VREMENSKOG DOMENA

Sažetak: Savremena astronomija vremenskog domena predstavlja naučnu oblast izrazito zasnovanu na obradi podataka. Podaci dolaze iz raznovrsnih izvora kao što su optički teleskopi različitog dizajna, spektrografi, detektori gravitacionih talasa, detektori neutrina, radio-prijemnici visoke rezolucije – kompleksni instrumenti koji se međusobno razlikuju po primarnoj naučnoj nameni. Podacima se u nekim slučajevima može pristupiti neposredno a nekad putem posredničkih servisā, koji se najčešće nazivaju brokerima. Pristup može imati oblik pretplate na tokove podataka u realnom vremenu ili ad-hoc upita nad arhivskim resursima. Takođe, više brokera može obrađivati iste podatke i pružati različite vrste ili nivoe njihovog obogaćivanja, od kojih svaki može biti naučno relevantan, dok kompozicija njihovih rezultata može imati još veću naučnu vrednost. Taj veliki ekosistem alata, tokova podataka i aplikacionih programskih interfejsa čini naučne podatke dostupnim na često neusklađen način, i to u smislu: načina pristupa podacima; strukture odgovora; identifikatora i semantičkih konvencija. Heterogenost na ovim različitim nivoima (sistemskom, strukturnom i semantičkom) je opšta pojava, a uspostavljanje međuzavisnosti između entiteta i prevazilaženje "vavilonske pometnje" predstavljaju osnovni naučni cilj.

Disertacija se bavi orkestracijom alata i heterogenih tokova podataka u astronomiji vremenskog domena. Cilj je predstavljanje sistema za izražavanje kompleksnih naučnih scenarija pomoću domenskih pojmova i njihovo izvršavanje nad raznorodnim servisima, uz normalizaciju značenja i strukture kao i uz očuvanje provenijencije dobijenih podataka. U radu su predstavljeni projektovanje i implementacija "nad-brokera" Alertissimo, softverskog prototipa opisanog sistema zasnovanog na razdvajanju semantičke i fizičke ravni domena. Doprinosi rada obuhvataju operativni semantički model i metamodel, razdvajanje naučne namere od fizičkog izvršenja, semantičku i strukturnu normalizaciju odgovora, planiranje izvršenja zasnovano na deklarisanim softverskim mogućnostima kao i kompoziciju i inkrementalno proširivanje tokova rada koji obuhvataju više brokera, uz očuvanje provenijencije fizičkih poziva. Rezultati potvrđuju primenljivost predloženog pristupa i pružaju osnovu za proširenje sistema ka obuhvatnoj multi-mesindžer orkestraciji i složenijim konstrukcijama tokova rada. Pored toga, DSL predstavlja dobru formalnu polaznu osnovu za obučavanje LLM modela namenjenog interfejsu zasnovanom na prirodnom jeziku.

Ključne reči: softverska orkestracija, domenski-specifični jezici, astronomija vremenskog domena, heterogeni tokovi podataka, deklarativna semantička interoperabilnost, upravljanje radnim tokovima, integracija podataka

Naučna oblast: Informacioni sistemi

Uža naučna oblast: Astroinformatika

SOFTWARE ORCHESTRATION OF HETEROGENEOUS DATA STREAMS AND PIPELINES IN TIME-DOMAIN ASTRONOMY

Abstract: Modern time-domain astronomy is a highly data-dependant scientific field. The data originates from various scientific instruments, such as distinct optical telescopes, spectrographs, gravitational wave / neutrino detectors, high-resolution radio receivers - all different in their design and observational purpose. The channels from which the data is accessed can be direct or mediated by value-added services, usually referred to by the community as brokers. Access can take the form of real-time subscription or ad-hoc queries against archival resources. Moreover, multiple brokers can operate on the same data and provide various types or levels of enrichment, all of which can be scientifically relevant – and composition of their outputs can be even more relevant. This huge ecosystem of tools, streams and APIs exposes scientific data through heterogeneous interfaces, response structures, identifiers, and semantic conventions – and yet establishing interdependencies between data entities is the core scientific goal.

This dissertation addresses the orchestration of tools and heterogeneous data streams in time-domain astronomy. Its objective is to enable scientific scenarios to be expressed in terms of domain concepts and subsequently executed across different services while preserving the meaning, structure, and provenance of the resulting data. The thesis presents the design and implementation of Alertissimo, an advanced research prototype based on a separation between the semantic and physical planes of the system. Contributions of the work include an operational semantic model and core metamodel, the separation of scientific intent from physical execution, declarative normalization of heterogeneous responses, capability-driven planning, and the composition and incremental extension of multi-broker workflows with preserved technical provenance. The results confirm the applicability of the proposed approach and provide a foundation for extending the system towards broader multi-messenger orchestration, more complex control flow, and long-running execution. Also, the DSL presents a good formal starting point for training an LLM to translate natural-language requests into the DSL, which would allow users to formulate complex scientific workflows without the need to learn the DSL syntax.

Key words: software orchestration, domain-specific language, time-domain astronomy, heterogeneous data streams, semantic interoperability, workflow management, data integration

Scientific field: Computer and information sciences

Scientific subfield: Astroinformatics

Sadržaj

1	Uvod.....	6
1.1	Teorijski kontekst i značaj teme.....	6
1.2	Predmet i cilj istraživanja.....	8
1.2.1	Polazne hipoteze istraživanja.....	10
1.3	Naučne metode istraživanja.....	12
1.4	Očekivani naučni doprinosi doktorske disertacije.....	12
1.5	Plan istraživanja i struktura rada.....	12
1.5.1	Plan istraživanja.....	12
2	Sistemi bazirani na događajima.....	14
2.1	Događaj (event).....	14
2.1.1	Jednostavan događaj (simple/raw event).....	15
2.1.2	Kompleksan događaj (complex event).....	16
2.2	Arhitektura sistema baziranih na događajima.....	16
2.2.1	Proizvođač (emiter) i konzument događaja.....	19
2.2.2	Kanal.....	20
2.2.3	Agent za obradu događaja.....	21
2.2.4	Konsultant.....	23
2.3	Ilustracija arhitekture primenjene na astronomiju.....	24
2.4	Jezici za obradu tokova događaja.....	25
2.4.1	Esper i Drools Fusion.....	25
2.4.1.1	Esper.....	25
2.4.1.2	Drools Fusion.....	26
2.4.1.3	Poređenje alata Esper i Drools.....	26
2.4.1.4	Primer Esper sintakse na astronomskom scenariju.....	27
	Osnove.....	27
	Prozori ili okviri (Windows).....	28
	Kontekst (Context).....	28
	Spajanje tokova (Joined Streams).....	29
	Obrasci (Patterns).....	29
3	Orkestracija procesa u sistemima za obradu događaja.....	31
3.1	Evolucija sistema za orkestraciju i kontrolu tokova procesa/rada.....	31
3.2	Uloga orkestratora.....	32
3.3	Evaluacija sinergije sistema za obradu događaja i orkestracije procesa	33

4	Astronomija vremenskog domena.....	35
4.1	Promenljivi astronomski objekti.....	35
4.1.1	Tranzijentni objekti.....	38
4.1.2	Događaji izazvani efektom gravitacionog sočiva.....	39
4.1.3	Promenljive zvezde.....	40
4.1.4	Planete u tranzitu.....	40
5	Klasifikacija astronomskih objekata u <i>BigData</i> eri.....	44
5.1	Karakter astronomskih podataka.....	46
5.2	Klasifikacija pomoću mašinskog učenja.....	47
5.3	Savremene metode klasifikacije za LSST/ZTF.....	48
5.3.1	Ansambl metode.....	48
5.3.2	Sekvencijalno duboko učenje.....	48
5.3.3	Transformer modeli za vremenske serije.....	49
5.3.4	Anomalije i otkrivanje retkih tipova događaja.....	49
5.3.5	Multi-modalno učenje.....	50
6	Ekosistem aplikacija vezanih za LSST/Rubin astronomski projekat.....	51
6.1	Uloga brokera u astronomiji vremenskog domena.....	51
6.2	Pregled LSST brokera.....	53
6.2.1	ALeRCE.....	53
6.2.2	AMPEL.....	54
6.2.3	ANTARES.....	54
6.2.4	Babamul/BOOM.....	55
6.2.5	Fink.....	55
6.2.6	Lasair.....	55
6.2.7	Pitt-Google.....	55
6.3	Poređenje LSST brokera.....	56
6.3.1	Koncepti dizajna.....	56
6.3.2	Funkcionalnosti.....	58
6.3.3	Performanse.....	59
6.4	Zaključak.....	59
7	Alertissimo: Koncept, namena i opšti pregled sistema.....	61
7.1	Motivacija za razvoj sistema.....	61
7.2	Osnovni pristup.....	63
7.3	Karakteristični uopšteni slučajevi korišćenja.....	65
7.3.1	Pronalaženje kandidata i obogaćivanje kod više brokera.....	65
7.3.2	Potvrđivanje kandidata nezavisnim izvorima.....	66
7.3.3	Povezivanje podataka iz različitih pregleda neba.....	66
7.3.4	Lokalno izdvajanje kandidata i nastavak toka rada.....	66
7.4	Arhitektura sistema.....	67

8	Ontologija i hijerarhijski semantički model.....	71
8.1	Uloga semantičkog sloja.....	73
8.2	Dvostruka uloga modela: struktura i ontologija.....	76
8.3	Organizacija semantičkog prostora.....	78
8.3.1	Kontejneri, polja i putanje.....	78
8.3.2	Parametarski kontejneri i dinamička polja.....	79
8.3.3	Ugnežđivanje.....	80
8.3.4	Tipovi i ograničenja.....	80
8.3.5	Semantički zapisi i kvalifikatori porekla.....	81
8.3.6	Semantički nivoi.....	82
8.3.7	Kompozicija.....	84
8.3.8	Korišćenje unutrašnjih projekcija.....	85
8.3.9	Kardinalnost.....	86
8.3.10	Semantičke veze.....	88
8.4	Portfolio.....	89
8.5	Normalizacija.....	91
8.6	Provenijencija.....	92
8.7	Standardi.....	93
8.8	Formalizacija.....	96
9	Sloj za orkestraciju i izvršenje.....	98
9.1	Granice odgovornosti orkestracionog sloja.....	99
9.2	WorkflowIR kao kanonska reprezentacija toka.....	100
9.3	Zajednički elementi IR koraka.....	101
9.4	Porodice semantičkih Step objekata.....	101
9.4.1	Razrešavanje identifikatora i otkrivanje kandidata.....	103
9.4.2	Pretraga i lokalno filtriranje.....	104
9.4.3	Pribavljanje postojećeg materijala i lokalno izvođenje.....	105
9.4.4	Povezivanje, ukrštanje i poređenje.....	105
9.4.5	Analiza, potvrđivanje, monitoring i akcije.....	106
9.5	Semantički predikati u IR-u.....	107
9.6	Validacija mogućnosti i izvršni graf.....	108
9.7	Planiranje fizičke realizacije.....	110
9.8	Pretvaranje kanonskih u fizičke parametre.....	111
9.9	Fizičko izvršenje i održavanje stanja.....	111
9.10	Normalizacija i kardinalnost rezultata.....	112
9.11	Akumulacija semantičkog materijala.....	114
9.12	Naknadno povezivanje koraka i identiteta kandidata.....	114
9.13	Ponovna upotreba fizičkog izvršenja.....	115
9.14	Inkrementalno proširivanje aktivnog toka.....	115

9.15	Rezultat, prikaz i provenijencija.....	116
9.16	Trenutni obuhvat i ograničenja izvršnog sloja.....	117
10	Sloj domenskog jezika - DSL.....	118
10.1	Uloga i projektni ciljevi DSL sloja.....	118
10.2	Od čitljive sintakse do proverljivog toka.....	119
10.3	Mesto DSL-a među semantičkim reprezentacijama.....	120
10.4	Sintaksa, gramatika i procesiranje DSL-a.....	121
10.4.1	Formalna gramatika i parsiranje.....	121
10.4.2	Pravila rasporeda i oblikovanja klauzula.....	122
10.5	Formiranje početnog skupa kandidata.....	122
10.5.1	Otkrivanje kandidata.....	123
10.5.2	Razrešavanje poznatih identifikatora.....	123
10.6	Prostorna, vremenska i količinska ograničenja.....	124
10.7	Semantički predikati.....	124
10.8	Razlika između where i filter.....	125
10.9	Pravci proširenja gramatike selekcije.....	126
10.10	Zahtevi za semantičkim obogaćivanjem.....	127
10.11	Potvrđivanje, povezivanje i prikaz rezultata.....	128
10.12	Površinski model, validacija i prevođenje u IR.....	129
10.13	Inkrementalni nastavak aktivnog toka rada.....	131
10.14	Javni programski interfejs DSL sloja.....	132
10.15	Celoviti primeri naučnih tokova.....	133
10.15.1	Višeopservaciono pronalaženje i poziciono povezivanje.....	133
10.15.2	Obogaćivanje preko više brokera.....	133
10.15.3	Lokalna odluka između udaljenih koraka.....	133
10.15.4	Potvrđivanje kvorumom.....	134
10.16	DSL kao proverljiva meta za NLP i vizuelne interfejse.....	134
10.17	Trenutni domet i ograničenja DSL sloja.....	135
11	Zaključak.....	138
11.1	Originalnost i doprinosi disertacije.....	138
11.2	Provera polaznih hipoteza.....	141
11.3	Ograničenja istraživanja.....	142
11.4	Pravci daljeg istraživanja i razvoja.....	144
12	Literatura.....	146
13	Biografija.....	153
13.1	Obrazovanje.....	153
13.2	Istraživački rad.....	153

13.3 Naučna saradnja.....	154
13.4 Bibliografija.....	154
Spisak objavljenih naučnih radova.....	156
Obrazac 10.....	163
Obrazac 11.....	164
Obrazac 12.....	165

1 Uvod

1.1 Teorijski kontekst i značaj teme

Savremena naučna istraživanja (ovo se naročito odnosi na fundamentalne nauke) sve manje mogu da se posmatraju odvojeno od softverskih sistema koji omogućavaju prikupljanje, integraciju, obradu i interpretaciju podataka. U disciplinama sa velikim obimom i brzinom nastajanja podataka softver više nije samo pomoćno sredstvo za automatizaciju pojedinačnih proračuna, već deo istraživačke infrastrukture kroz koji se operacionalizuju metodologija, izbor podataka i redosled analitičkih postupaka. Paradigma data-intenzivne nauke, često označena kao „četvrta paradigma“ naučnog istraživanja, upravo naglašava ovu promenu: otkriće sve češće zavisi od sposobnosti da se veliki i heterogeni skupovi podataka sistematski povežu sa računarskim metodama i servisima (Hey et al., 2009).

Iz perspektive softverskog inženjerstva, takvo okruženje prirodno vodi ka sistemima sastavljenim od većeg broja autonomnih komponenti koje se razvijaju, menjaju i skaliraju nezavisno. Klasični principi modularnosti i skrivanja implementacionih odluka pokazuju da održivost složenog sistema zavisi od načina na koji su granice između komponenti postavljene, a ne samo od njihovih pojedinačnih funkcionalnosti (Parnas, 1972). Na nivou softverske arhitekture isti problem se iskazuje kroz izbor stabilnih apstrakcija, jasno definisanih interfejsa i obrazaca saradnje između komponenti, čime se umanjuje zavisnost sistema od konkretnih tehnoloških realizacija (Shaw & Garlan, 1996). Za naučnu infrastrukturu, u kojoj se postojeći alati i izvori podataka ne mogu centralno kontrolisati niti istovremeno standardizovati, ovakva modularnost predstavlja preduslov za dugoročnu evolutivnost sistema.

Poseban problem nastaje kada komponente ne dele isti model podataka, terminologiju ili način pristupa. Integracija tada ne rešava samo pitanje prenosa ili konverzije formata, već posredovanja između različitih reprezentacija istog ili srodnog domena. Koncept softverskih medijatora uveden je upravo kao arhitekturni sloj koji razdvaja aplikacionu logiku od heterogenih izvora i obavlja apstrakciju, spajanje i tumačenje podataka (Wiederhold, 1992). U savremenom naučnom okruženju ovaj zahtev se dodatno povezuje sa principima mašinski čitljive interoperabilnosti, eksplicitne semantike i porekla podataka; FAIR principi, na primer, interoperabilnost neposredno vezuju za formalne i zajedničke jezike reprezentacije znanja i za očuvanje provenijencije (Wilkinson et al.,

2016).

Drugi važan softverski koncept jesu naučni radni tokovi (engl. *scientific workflows*). Oni omogućavaju da se složena analiza predstavi kao kompozicija koraka koji mogu uključivati pristup bazama podataka, izvršavanje algoritama, pozive udaljenih servisa, transformaciju podataka i računarski intenzivne poslove. Sistemi kao što je "Kepler" pokazali su da naučni tok može biti eksplicitno modelovan nezavisno od infrastrukture na kojoj se pojedini koraci izvršavaju (Ludäscher et al., 2006). Šira istraživanja naučnih radnih tokova izdvajaju kompoziciju, planiranje i izvršavanje, upravljanje zavisnostima, praćenje stanja, ponovljivost i provenijenciju kao ključne probleme ove klase sistema (Gil et al., 2007; Deelman et al., 2009). U tom smislu, orkestracija predstavlja sloj koji povezuje deklarisanu nameru korisnika sa konkretnim redosledom i načinom izvršavanja raspoloživih komponenti.

Za smanjenje zavisnosti između naučne namere i implementacionih detalja posebno su značajni deklarativni pristupi. Umesto da korisnik propisuje konkretan API poziv, naziv fizičkog parametra ili format odgovora, sistemu se može zadati šta se želi dobiti ili koji uslov treba da bude zadovoljen, dok se izbor i povezivanje konkretnih mehanizama prepušta sloju za planiranje i izvršavanje. Domenski-specifični jezici (DSL) predstavljaju jedan od standardnih načina da se takve apstrakcije izraze terminima bliskim korisničkom domenu, uz ograničen i formalizovan jezik namenjen određenoj klasi problema (Fowler, 2010). Ako se isti princip primeni i na podatke, deklarativna semantička mapiranja mogu da opišu vezu između fizičkih polja pojedinačnih izvora i zajedničkog domenskog modela, dok generički transformacioni mehanizam obavlja njihovo prevođenje. Time se interoperabilnost ne zasniva na nizu *ad-hoc* konverzija ugrađenih u poslovnu ili naučnu logiku, već na eksplicitnim, proverljivim deklaracijama.

Ovakva arhitektura je naročito relevantna za sisteme koji istovremeno kombinuju tokove događaja i servisno orijentisane ili interfejse zasnovane na obrascu zahtev–odgovor. Kontinuirani tokovi događaja zahtevaju pravovremenu reakciju i obradu pod vremenskim ograničenjima, dok dodatni kontekst često mora biti pribavljen pozivima ka spoljnim servisima ili bazama podataka. Orkestrator zato ne zamenjuje same specijalizovane komponente: njegova uloga je da nad njima uspostavi upravljački sloj koji povezuje semantički opis zadatka, raspoložive sposobnosti sistema, zavisnosti između koraka i stanje izvršavanja. Detaljniji koncepti arhitektura baziranih na događajima i orkestracije obrađeni su u narednim poglavljima.

Sa stanovišta softverskog inženjerstva, osnovni problem koji prati razvoj savremene astronomije nije samo velika količina podataka, već i njihova izrazita heterogenost. Različiti teleskopi, brokeri i naučni servisi podatke predstavljaju na različite načine, koriste različite programske interfejse i nude različite mogućnosti obrade. Naučni scenario koji zahteva podatke ili rezultate iz više takvih izvora zato lako postaje zavisn od konkretnih tehnologija kojima se pristupa svakom pojedinačnom sistemu. Jedan od mogućih odgovora na ovaj problem jeste uvođenje zajedničkog semantičkog nivoa na kojem se podaci i naučne operacije opisuju nezavisno od svojih fizičkih reprezentacija. Takav pristup omogućava da se heterogeni podaci prevode u zajednički model, da se naučni postupci predstavljaju kao niz semantički definisanih koraka, a da se tek naknadno određuje kojim će servisima i na koji način ti koraci biti izvršeni. Naučni scenario se na najvišem nivou može opisati deklarativno, u terminima samog domena, dok sistem preuzima odgovornost za njegovo prevođenje u konkretne pozive, povezivanje međurezultata i očuvanje informacija o njihovom poreklu. Na taj način se postiže veći stepen interoperabilnosti i smanjuje neposredna zavisnost naučnih tokova od pojedinačnih formata podataka, API-ja i mesta izvršenja. Formalizovanje domena kroz zajedničku semantiku i deklarativni jezik istovremeno stvara mašinski čitljivu osnovu na kojoj se, kao prirodna nadogradnja, mogu graditi interfejsi zasnovani na obradi prirodnog jezika i velikim jezičkim modelima: AI sloj tada ne određuje izvršenje neposredno, već korisničku nameru prevodi u formalnu i proverljivu reprezentaciju sistema.

1.2 Predmet i cilj istraživanja

Astronomija vremenskog domena proučava objekte i pojave čije se osobine menjaju na vremenskim skalama od delova sekunde do više godina. U ovu oblast spadaju, između ostalog, supernove, aktivna galaktička jezgra, promenljive zvezde, mikrolensing događaji i drugi tranzijentni ili promenljivi fenomeni. Za značajan deo ovih pojava naučna vrednost zavisi od pravovremene detekcije, klasifikacije i pribavljanja dodatnog konteksta, zbog čega se podaci ne mogu posmatrati samo kao statičke kolekcije već i kao tokovi događaja nad kojima se odluke donose u ograničenom vremenu.

Pritom savremena astronomija sve više kombinuje raznorodne instrumente i izvore podataka. Optičkim, infracrvenim, radio i visokoenergetskim opservatorijama pridružili su se detektori gravitacionih talasa i neutrinske opservatorije, pa se isti astrofizički događaj može istraživati kroz više međusobno komplementarnih kanala (Abbott et al., 2016; Mészáros et al., 2019). Ovakav

multi-mesindžer pristup podrazumeva koordinaciju signala zabeleženih različitim instrumentima o istom ili potencijalno povezanom astronomskom događaju. Instrumenti i servisi koji učestvuju u takvom ekosistemu geografski su i organizaciono razdvojeni, razvijani su u različitim tehnološkim okruženjima i koriste različite modele podataka i interfejsa.

Skala problema posebno je vidljiva kod velikih preglednih projekata. Vera C. Rubin Observatory i program Legacy Survey of Space and Time (LSST) projektovani su za sistematsko i učestalo snimanje velikog dela neba i generisanje miliona obaveštenja (engl. *alerts*) o promenama tokom jedne posmatračke noći, uz ukupne skupove podataka koji se mere desetinama petabajta (LSST Science Collaboration, 2009; Ivezić et al., 2019; Graham et al., 2024). Pored LSST-a, vremenski promenljive fenomene prate Zwicky Transient Facility (ZTF), ATLAS, BlackGEM i drugi zemaljski projekti, kao i svemirske misije Euclid i Nancy Grace Roman. General Coordinates Network (GCN) istovremeno distribuira obaveštenja o događajima sa instrumenata kao što su LIGO/Virgo/KAGRA, IceCube, Swift i Fermi. Rezultat nije jedan homogen tok, već skup vremenski zavisnih, tehnički i semantički različitih izvora koji se moraju korelisati u skladu sa konkretnim naučnim ciljem.

U LSST ekosistemu dodatni sloj čine posrednički sistemi koji se u astronomskoj zajednici nazivaju brokerima. Brokeri primaju alert tokove, obogaćuju ih kontekstom, primenjuju filtre i klasifikatore, omogućavaju pretragu istorijskih podataka i nude različite oblike kros-meča sa katalogima.

ALeRCE, AMPEL, ANTARES, Fink, Lasair i drugi brokeri dele deo funkcionalnosti, ali se razlikuju u dostupnim proizvodima, API interfejsima, parametrima, strukturama odgovora, klasifikatorima i modelima provenijencije (Förster et al., 2021; Möller et al., 2021; Nordin et al., 2019; Vujčić et al., 2025). IVOA standardi, uključujući VOEvent, značajno doprinose interoperabilnosti astronomskih podataka (Seaman et al., 2011), ali u praksi ne uklanjaju heterogenost operativnih interfejsa i semantičkih reprezentacija koje nude nezavisni servisi.

Predmet istraživanja ove doktorske disertacije jeste upravo problem softverske orkestracije takvog heterogenog ekosistema: kako naučni scenario izraziti u terminima domena, a zatim ga izvršiti preko više autonomnih izvora i alata bez ugrađivanja znanja o svakom konkretnom interfejsu u samu naučnu logiku. Poseban aspekt predmeta istraživanja čini deklarativna semantička interoperabilnost, odnosno mogućnost da se veze između specifičnih struktura podataka iz različitih izvora i zajedničkih astronomskih koncepata opišu mapiranjima, dok generički sloj obavlja njihovu transformaciju u kanonsku reprezentaciju. Na isti način, fizički zahtevi pojedinačnih servisa treba

da ostanu odvojeni od semantičke namere korisnika i da budu rešavani u fazama planiranja, vezivanja i izvršavanja.

Cilj istraživanja je projektovanje, implementacija i evaluacija prototipa sistema Alertissimo za orkestraciju alata i tokova podataka u astronomiji vremenskog domena. Predloženi sistem treba da obuhvati: zajednički hijerarhijski semantički model astronomskih podataka; ontologiju i međureprezentaciju orkestracionih koraka nezavisnu od konkretnih provajdera; deklarativna mapiranja fizičkih podataka u kanonske semantičke strukture; domenski-specifičan jezik za formulisanje naučnih scenarija; planiranje i izvršavanje tih scenarija nad heterogenim servisima; kao i očuvanje provenijencije i sledivosti kroz više faza obrade. Time prototip služi i kao eksperimentalna platforma za proveru hipoteze da se heterogeni brokeri i drugi izvori mogu komponovati na višem nivou apstrakcije, bez njihove prethodne uniformizacije.

Domenski-specifični jezik u ovom radu ima ulogu deklarativnog korisničkog sloja: naučnik opisuje željeni skup objekata, potrebne vrste podataka, uslove filtriranja, povezivanja ili analize, dok sistem određuje kako se takva namera može realizovati raspoloživim sposobnostima konkretnih provajdera. DSL i ontologija su pri tome namerno formalizovani i kao stabilna, mašinski čitljiva osnova za naredni nivo korisničke interakcije. Prirodna nadogradnja sistema zato mogu biti NLP/LLM interfejsi koji mogu slobodno formulisan naučni zahtev prevoditi u formalni DSL i/ili semantičku međureprezentaciju, uz proveru prema definisanoj semantici pre planiranja i izvršenja. Vizuelni alati mogu koristiti isto formalno jezgro kao komplementaran način definisanja toka. Tako je cilj rada istovremeno domenski - omogućavanje složenih scenarija u astronomiji vremenskog domena - i softversko-inženjerski - razvoj i validacija arhitekturnog pristupa za orkestraciju, semantičku integraciju i deklarativno upravljanje heterogenim naučnim servisima.

1.2.1 Polazne hipoteze istraživanja

Na osnovu analize dostupne, relevantne literature, i postavljenog predmeta i cilja disertacije mogu se postaviti opšte i posebne hipoteze.

Opšte hipoteze:

H(0.1) Heterogeni tokovi podataka ili alati sa specifičnim funkcijama se mogu orkestrirati prema potrebama naučnih scenarija za otkrivanje i svrstavanje promenljivih astronomskih

objekata i fenomena.

- H(0.2) Tokovi podataka koji se emituju sa savremenih astronomskih teleskopa i alati za obradu tokova su ključni činioci astronomskog softverskog ekosistema i doprineće ne samo velikom kvantitativnom porastu otkrića naučnih fenomena već i kvalitativnim pomacima u istraživačkim metodama kao i demokratizaciji nauke i resursa.

Posebne hipoteze:

- H(0.1.1) Moguće je formirati ontologiju generalizovanih gradivnih blokova (npr. kros-mečer kataloga, fotometrijska klasifikacija itd) koji se mogu nezavisno razvijati, a zatim orkestrirati u složene tokove prilagođene specifičnim naučnim slučajevima (npr. potraga za supernovama tipa Ia, identifikacija kandidata za praćenje spektroskopijom).
- H(0.1.2) Orkestrirani sistemi mogu dinamički menjati tok obrade i uključivati dodatne izvore podataka ili alate na osnovu međurezultata, čime se optimizuje korišćenje resursa i vreme reakcije.
- H(0.1.3) Da bi orkestracija heterogenih tokova i alata bila moguća u okviru proizvoljnih naučnih scenarija, neophodno je definisati zajednički semantički model podataka i servisa koji uspostavlja vezu između generalizovanih koncepata domena i konkretnih implementacionih interfejsa.
- H(0.1.4) Potrebno je uvođenje domenski-specifičnog jezika za deklarativno definisanje naučnih scenarija i izvršavanje orkestracionih tokova, koji bi formalizovao domenske koncepte na višem nivou apstrakcije i smanjio zavisnost od konkretnih implementacija i tehnologija.
- H(0.2.1) Povećanje obima i kontinuiteta tokova podataka iz savremenih teleskopa dovodi do značajnog rasta broja detektovanih astronomskih fenomena, posebno u domenu retkih i kratkotrajnih događaja koji moraju biti prioritetno ispitani u ograničenom vremenskom roku
- H(0.2.2) Dostupnost vremenski gustih i dugotrajnih nizova fotometrijskih posmatranja iz više izvora, uz dodatne vrste podataka poput isečaka slike i spektroskopije, omogućava multimodalnu karakterizaciju astronomskih objekata i fenomena, čime se znatno unapređuje kvalitet naučnih interpretacija.

1.3 Naučne metode istraživanja

Osnovne metode koje su se koristile za realizovanje istraživanja su:

- Prikupljanje i sistematizacija podataka o dostupnim rešenjima na osnovu literature, razmenom informacija i neposrednim kontaktima sa relevantnim naučnim subjektima u zemlji i svetu
- Kritička analiza postojećih rešenja
- Postavljanje sopstvene metodologije za ostvarivanje postavljenih ciljeva istraživanja sintezom poznatih i proverenih elemenata rešenja, ili njihovom modifikacijom, sa originalnim elementima rešenja
- Eksperimentalna provera predložene metodologije izradom prototipa
- Praktična provera postavljenih hipoteza testiranjem realizovanog prototipa

1.4 Očekivani naučni doprinosi doktorske disertacije

Očekivani naučni doprinosi istraživanja su:

- Pregled i komparacija postojećih pristupa za manipulaciju i analizu tokova podataka astronomije vremenskog domena
- Definisanje generalizovanog hijerarhijskog semantičkog modela domena astronomskih tranzijenata
- Definisanje ontološkog modela orkestracije alata i tokova podataka
- Definisanje DSL-a (domenski specifičnog jezika) za formulisanje naučnih scenarija nad ovako definisanom arhitekturom.
- Implementacija prototipa softvera za orkestraciju heterogenih tokova podataka u astronomiji vremenskog domena

1.5 Plan istraživanja i struktura rada

1.5.1 Plan istraživanja

U cilju izrade i odbrane doktorske disertacije, planirane su sledeće aktivnosti:

- Prikupljanje, analiza i sistematizacija postojećih naučnih rezultata iz oblasti astronomije vremenskog domena, obrade tokova podataka i softverskih sistema za njihovu analizu
- Analiza postojećih pristupa modelovanju i integraciji heterogenih tokova astronomskih podataka, sa posebnim osvrtom na ograničenja u pogledu interoperabilnosti i kompozicije
- Formulisanje problema jedinstvenog semantičkog predstavljanja podataka iz različitih izvora (teleskopi, brokeri, katalozi) u kontekstu astronomskih tokova
- Definisanje semantičkog modela podataka koji omogućava unifikaciju ključnih koncepata (detekcije, objekti, svetlosne krive, kros-mečeri) uz očuvanje informacija o poreklu i strukturi podataka
- Definisanje domen-specifičnog jezika (DSL) zasnovanog na semantičkom modelu, namenjenog deklarativnom opisu upita i orkestracije nad heterogenim tokovima podataka
- Razvoj prototipa sistema koji implementira predloženi semantički model i DSL, uključujući podršku za integraciju više izvora podataka i alata
- Validacija predloženog pristupa kroz primenu na reprezentativnim naučnim scenarijima iz astronomije vremenskog domena (npr. identifikacija specifičnih klasa objekata ili fenomena)
- Analiza rezultata, provera postavljenih hipoteza i izvođenje zaključaka o efikasnosti i primenljivosti predloženog pristupa

2 Sistemi bazirani na događajima

Drugi deo ovog rada uvodi teoretske koncepte arhitekture sistema baziranih na događajima i principe orkestracije softverskih komponenti kao osnovu za predloženo rešenje i dalje istraživanje.

Softverski sistem koji je baziran na događajima (engl. *event driven*) okrenut je ka percepciji nečeg značajnog što dolazi iz spoljnog sveta, kako god ga mi definisali. Može se napraviti jednostavna analogija sa načinom na koji funkcioniše ljudski organizam: različita čula primaju informacije, one se prenose putem nervnog sistema, mozak ih obrađuje i donosi zaključak na osnovu kog mišići reaguju (tabela 1).

Uloga	Event-driven architecture	Ljudsko telo
<i>Interakcija sa okruženjem</i>	Fizički senzori, log fajlovi, transakcije, mehanizmi za detekciju...	Čula
<i>Prenos informacija između senzora i mehanizama za zaključivanje</i>	Sistemi za razmenu poruka	Nervni sistem
<i>Procesiranje informacija, učitavanje konteksta, razumevanje okoline, primena paterna na osnovu pravila ili sličnih istorijskih iskustava, formulacija reakcije</i>	Sistemi za zaključivanje, neuronske mreže, Bayesove mreže, statistika, analitika, semantika...	Mozak
<i>Egzekucija reakcije</i>	Konzumenti događaja, trigeri	Mišići i refleksi

*Tabela 1: Ljudsko telo kao "event-driven" sistem,
bazirano na (Wiener, 1948)*

2.1 Događaj (event)

Događaj zauzima centralno mesto u event-driven sistemu. U okviru takvog sistema, događaj je:

"Diskretna pojava unutar definisanog domena

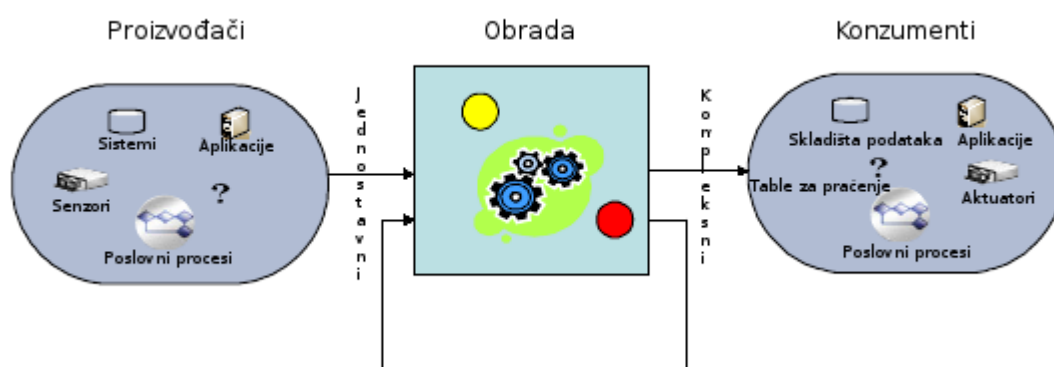
koju je sistem sposoban da detektuje"

Pod **diskretnom pojavom** podrazumevamo informaciju dobijenu u definisanom vremenskom trenutku koja se ne može promeniti. Događaj sugerise da se nešto već desilo, a ne želju da se nešto desi (kao u request-response sistemima). U karakteru event-driven sistema je diskretizacija događaja, bilo da je priroda same pojave diskretna ili kontinualna.

Domenom se definiše skup pojava od interesa za konkretni sistem. Time se sužava perceptivna moć sistema na praćenje stvari od značaja.

Sposobnost detekcije se odnosi na tehničke kapacitete senzora i svest o detektabilnosti određenih pojava. Za uspešan sistem neophodan je sklad ove dve stvari.

Ključna funkcija event-driven sistema je izvođenje kompleksnih događaja na osnovu niza jednostavnih događaja. Na slici 1 dat je uprošćen prikaz ovakve transformacije.



Slika 1: Od jednostavnih do složenih događaja. Prilagođeno prema konceptima iz knjige "Event Processing in Action" Etzion & Niblett 2010.

2.1.1 Jednostavan događaj (simple/raw event)

Jednostavan (ili sirov) događaj dešava se na ivici sistema i biva opažen perceptivnim mehanizmima sistema.

Jednostavan događaj je događaj koji se na izabranom nivou posmatra kao nedeljiv. Granulacija se definiše na nivou sistema i zavisi od karaktera senzora. Na primer, sistem za praćenje avio saobraćaja može preko radara primiti jednostavan događaj da je avion sleteo, dok se u unutrašnjem sistemu za praćenje rada aviona sletanje posmatra kao kompleksan događaj i izveden je iz niza jednostavnih događaja.

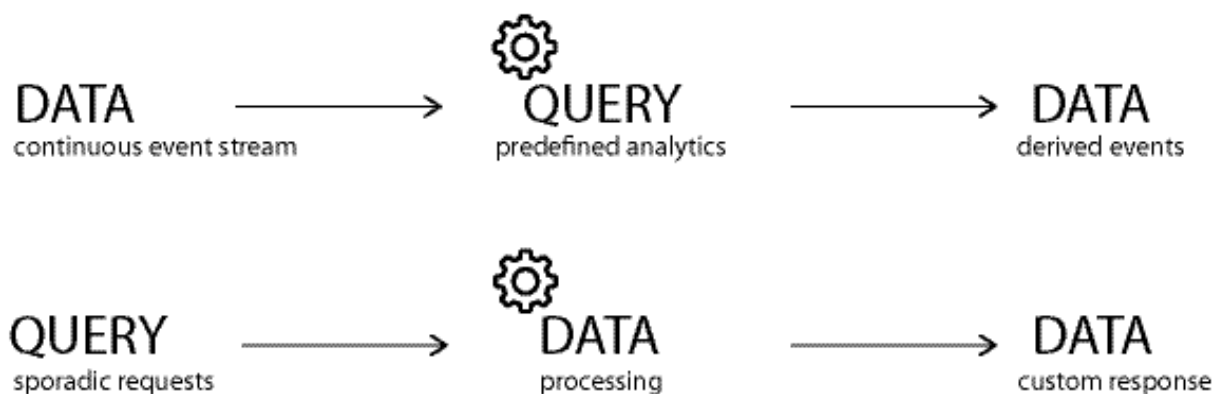
2.1.2 Kompleksan događaj (complex event)

Kada kažemo da se desio kompleksan događaj to je plod zaključivanja nad skupom jednostavnih događaja. Kompleksan događaj je apstrakcija jednostavnih događaja koje različitim mehanizmima "sumira, označava i reprezentuje" (Luckham, 2011).

2.2 Arhitektura sistema baziranih na događajima

Arhitektura sistema baziranih na događajima (engl. *event-driven architecture*), koju neki autori nazivaju i programiranje zasnovano na događajima (engl. *event-based programming*) podrazumeva "stil softverske arhitekture u okviru kojeg jedna ili više komponenti reaguju na notifikacije o jednom ili više događaja" (Etzion & Niblett, 2010). Ovakav sistem orkestrira ponašanje između proizvodnje, detekcije i konzumpcije događaja.

Razlika u paradigmi sistema baziranih na događajima i aplikacija koje se baziraju na zahtevu i odgovoru (engl. *request-response*) nalazi se u prirodi upita. Dok se kod klasičnih sistema izgrađenih na relacionim bazama podataka ili kod sistema agregiranih podataka upit povremeno upućuje prema sačuvanim podacima, kod event-driven arhitekture sveži podaci kontinuirano prolaze kroz upite (Tabela 2). Razlika sa sistemima povremene (engl. *batch*) obrade je da se proces detekcije obavlja neprestano, a ne u zakazanim intervalima.



Slika 2: Razlika u obradi podataka između sistema zasnovanih na događajima i tradicionalnih transakcionih sistema (Vujčić & Jevremović, 2020)

Opređenjenje ka *event-based* pristupu može biti primamljivo u sledećim slučajevima (Etzion & Niblett, 2010):

- Kada je priroda sistema prirodno okrenuta ka događajima
- Kada postoji potreba za identifikacijom i reakcijom na određenu situaciju u momentu kada se ona desila.
- Kada logika za obradu događaja treba da se odvoji od ostatka aplikacije. Ovo omogućava brzu adaptaciju novim zahtevima.
- Kada aplikacija zahteva analizu velikih količina podataka. Podaci se mogu organizovati u tokove ("streams") i distribuirati na više nodova radi paralelizovane obrade.

	Transakcije i analitičke aplikacije	Event-based aplikacije
<i>Paradigma upita</i>	Ad-hoc zahtevi ili upiti	Neprekidan tok upita

<i>Latencija odziva sistema</i>	Niska za konzistentne sisteme	Niska ali nestalna za reaktivne sisteme
<i>Skaliranje i otkazi</i>	Monolitni	Odvojeni
<i>Volumen zahteva</i>	Visok, konzistentan	Ekstremno visok, manje konzistentan

Tabela 2: Razlike između tradicionalnih request/response aplikacija i event-based aplikacija

Arhitekturu sistema baziranog na događajima treba da odlikuju svojstva interoperabilnosti, distribuiranosti, modularnosti, apstrakcije i elastične povezanosti (engl. *loose coupling*). Neke od karakteristika ovakvog sistema su:

- *Udaljeni delovi sistema ne treba da znaju jedni za druge.* Ovo naročito važi za delove na ivicama, kao što su emiteri i konzumenti događaja, koji ne treba da imaju izravni dodir.
- *Komunikacija se izoluje kroz odvojene kanale.* Time se smanjuje kompleksnost i postiže modularnost
- *Tehnologija implementacije delova sistema može biti heterogena.* Ovo je fundamentalni zahtev, bilo da se posmatra sa strane adaptacije postojećeg stanja sistema ili anticipacije budućeg.
- *Poruke koje se razmenjuju u okviru sistema treba da su u standardizovanom obliku.* Ovakva standardizacija je poželjna ali često neostvariva jer se u praksi traže izvodljiva rešenja.
- *Jezik kojim se opisuju mehanizmi zaključivanja je na visokom nivou apstrakcije.* Koncepti kompleksne obrade događaja treba da budu inherentno ugrađeni u takav jezik.

Ovako elastično povezan sistem omogućava asinhronu komunikaciju, nezavisno skaliranje i povećanu otpornost na otkaze njegovih pojedinačnih delova.

U sledećem delu predstavimo "gradivne blokove" ovakve arhitekture.

2.2.1 Proizvođač (emiter) i konzument događaja

Proizvođač događaja (engl. event producer, event emitter) je entitet na ivici sistema koji uvodi događaje u sistem.

Koncept proizvođača događaja se može predstaviti na nekoliko načina (Etzion & Niblett, 2010): to može biti apstraktni tip, skup instanci istog tipa ili jedna instanca – što zavisi od homogenosti i unikatnosti proizvođača. Npr, ne bi se na isti način modelovao sistem u kojem emitere predstavljaju hiljade identičnih GPS senzora i sistem u kojem su proizvođači nekoliko različitih teleskopa, sa heterogenom strukturom podataka.

Navešćemo nekoliko primera proizvođača događaja:

- **Hardverski senzori** – beleže jednu ili više fizičkih karakteristika iz okruženja (temperatura, vlažnost, pokret, pritisak itd)
 - RFID (identifikacija pomoću radio frekvencija) čitač
 - Seizmometar
 - GPS lokator
 - Optički teleskop
- **Softverski proizvođači**
 - Simulatori hardverskih senzora (u kasnijim poglavljima videti: AlertSim)
 - Monitori rada aplikacija ili mreža
 - Agregatori spoljnih tokova podataka

Konzument događaja (engl. event consumer, event sink) je entitet na ivici sistema koji prima obrađene događaje. Uloga konzumenta može biti automatizovana reakcija na događaje, obaveštavanje o događajima, predstavljanje događaja, arhiviranje itd.

Nekoliko primera konzumenata:

- **Hardverski aktuatori** – automatizovano obavljaju mehaničku akciju pomoću hidrauličkog

/ pneumatskog / električnog / magnetnog (i dr.) sistema, kao što je paljenje ili gašenje svetla, kočenje na vozilu itd.

- **Obaveštavanje i predstavljanje** – obavljaju se u interakciji sa čovekom, putem kanala za komunikaciju i softverskih aplikacija. Obaveštavanje je jednostavno – alarm, slanje emaila ili sms poruke, notifikacija preko socijalnih mreža. Predstavljanje se odnosi na obradu podataka i vizuelizaciju na softverskim "tablama za praćenje" (engl. dashboard).
- **Softverski konzumenti** – sistemi za čuvanje podataka (npr. baze podataka), sistemi za upravljanje poslovnim procesima itd

2.2.2 Kanal

Kanal služi za distribuiranje događaja između proizvođača, konzumenata i agenata unutar sistema baziranog na događajima. Koncept kanala se uvodi da bi se dobilo na modularnosti i elastičnijoj povezanosti komponenti unutar sistema.

Emiter događaja delegira distribuciju događaja kanalu, u kojem se izoluje logika koja je vezana za razmenu poruka sa događajima, pretplaćivanje korisnika, tehnički aspekt komunikacije itd. U implementaciji, kanal može poprimiti različite oblike – od specijalizovanog softverskog modula, preko multicast¹ protokola do postojećih softverskih okvira za razmenu poruka (kao što je Apache Kafka, JMS – Java Messaging Service) ili web platformi za razmenu poruka (Pusher, PubNub i mnogi drugi).

Odlike ovakvog koncepta kanala su:

- Omogućavanje protoka događaja između n proizvođača/agenata i m agenata/konzumenata.
- Razdvajanje proizvođača, agenata i konzumenata.
- Redukcija $m \times n$ broja veza što olakšava održavanje i dijagnostiku
- Nikada se ne menja sam događaj, samo odluke o njegovoj distribuciji
- Primenjuje paterne poput publish/subscribe

¹ Tehnika za transmisiju podataka od jednog emitera ka više primalaca odjednom

- Jasnoća pri modelovanju u slučaju puno veza između modula

2.2.3 Agent za obradu događaja

Agent za obradu događaja (engl. *event processing agent*, dalje: EPA) je modul koji primenjuje razne operacije nad ulaznim (jednostavnim) događajima u cilju donošenja zaključka o kompleksnom događaju. EPA modul treba da primeni ekspertsku i statistička znanja nad podacima.

Ključne uloge EPA su **filtriranje**, **transformacija** i **detekcija paterna**.

Filtriranje spada u jednostavnu obradu. Ovde ne dolazi do modifikacije događaja kao ni do zaključivanja. Cilj filtriranja je da se smanji obim događaja od interesa pre primene mehanizama zaključivanja.

Transformacija je promena vrednosti ili strukture događaja u odnosu na definisanu funkciju derivacije. Načini transformacije mogu biti (Etzion & Niblett, 2010):

- *prevođenje* – primena funkcije derivacije nad ulazom
- *sastavljanje* – spajanje događaja iz različitih izvora
- *deljenje* – kreiranje niza događaja od jednog događaja
- *agregacija* – primena funkcije agregacije (AVG, MAX, COUNT...) nad nizom događaja
- *obogaćivanje* – dodavanje podataka iz spoljnih izvora
- *projekcija* - biranje podskupa atributa

Detekcija paterna podrazumeva sposobnost prepoznavanja kompleksnog događaja na osnovu vrednosti atributa jednog ili više događaja ili tipova događaja u određenom kontekstu. Paterni se dele na osnovne i dimenzionalne^(Etzion & Niblett, 2010):

- **Osnovni paterni**
 - Logički paterni (svi, neki, nijedan)

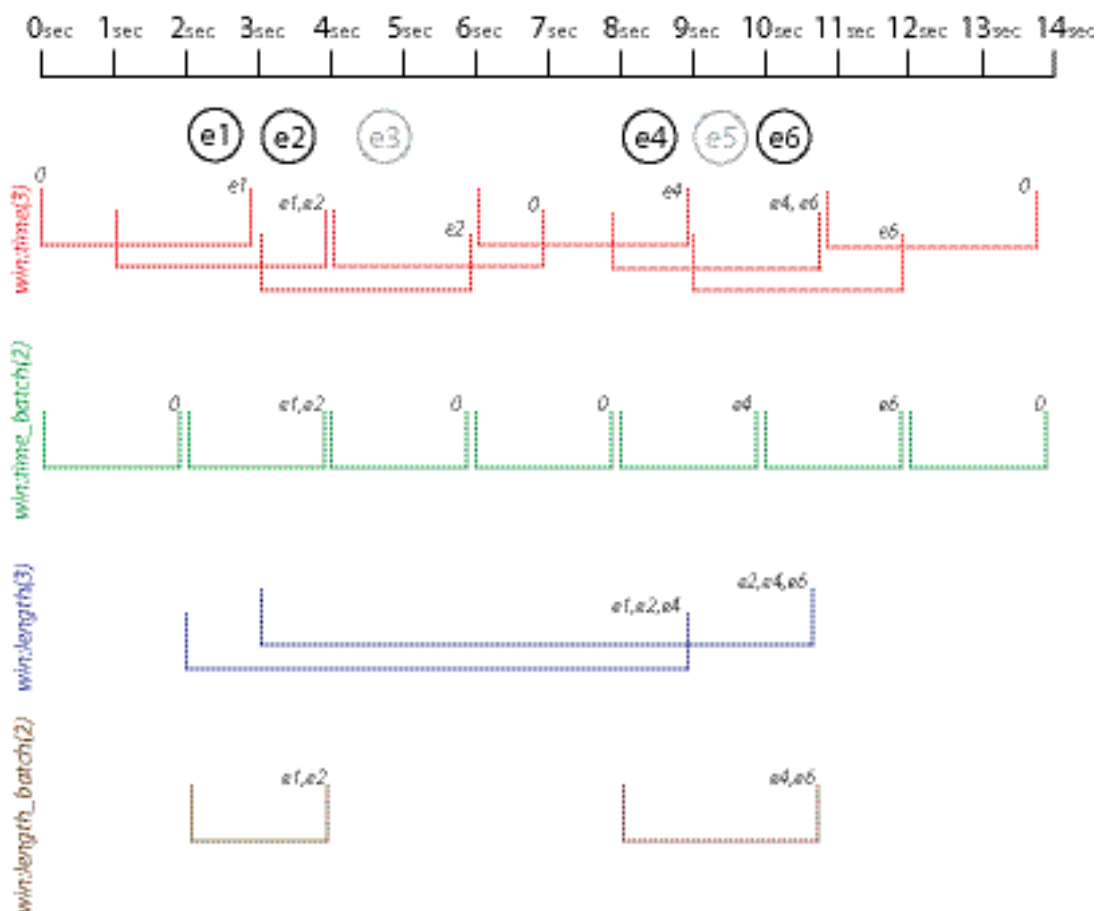
- Paterni praga (prosek, min, max, zbir, brojanje)
- Paterni selekcije podskupa (n-najvećih, n-najmanjih)
- Modalni paterni (uvek, nikad, ponekad)
- **Dimenzionalni paterni**
 - Paterni redosleda u vremenu (prvih n, poslednjih n, sekvenca – uključujući vremensku međuzavisnost događaja primenom intervalne algebre² (Allen, 1983) (pre, posle, za vreme, preklapanje itd.))
 - Paterni trenda u vremenu (povećanje, smanjenje, stabilnost, nerastući trend, neopadajući trend, kombinovani trend)
 - Paterni prostorne distance (minimalna distanca, maksimalna distanca, prosečna distanca...)
 - Paterni kretanja / vremensko-prostorni paterni (kretanje u konstantnom pravcu, kretanje ka, nekretanje...)

Kontekst može biti:

- *vremenski* – baziran na vremenskom rasporedu događaja. Može se posmatrati u odnosu na određeni interval ili na broj događaja u vremenu
- *prostorni* – u odnosu na prostornu poziciju događaja
- baziran na stanju sistema u određenom trenutku

Kontekst se uvodi pomoću tzv. okvira (engl. *Windows*).

2 Allen's interval algebra: <http://www.ics.uci.edu/~alspaugh/cls/shr/allen.html>



Slika 3: Ilustracija vremenskih okvira u obradi događaja

U međunarodnoj astronomskoj zajednici, specifično u njenom delu koji se bavi promenljivim događajima, ustalio se termin *broker* za softverske alate koji uzimaju EPA ulogu u sistemu obrade događaja. Ovaj termin nije sasvim u skladu sa formalnom literaturom gde se kao broker obično navodi modul koji ima više logističku i posredničku ulogu (npr. Apache Kafka³) a manje na usko specifičnu domenski-analitičku. Uprkos tome, zbog opšteprihvaćenosti termina u zajednici, u daljem radu referisaćemo se na broker kao na EPA agent u ekosistemu astronomskih tokova podataka.

2.2.4 Konsultant

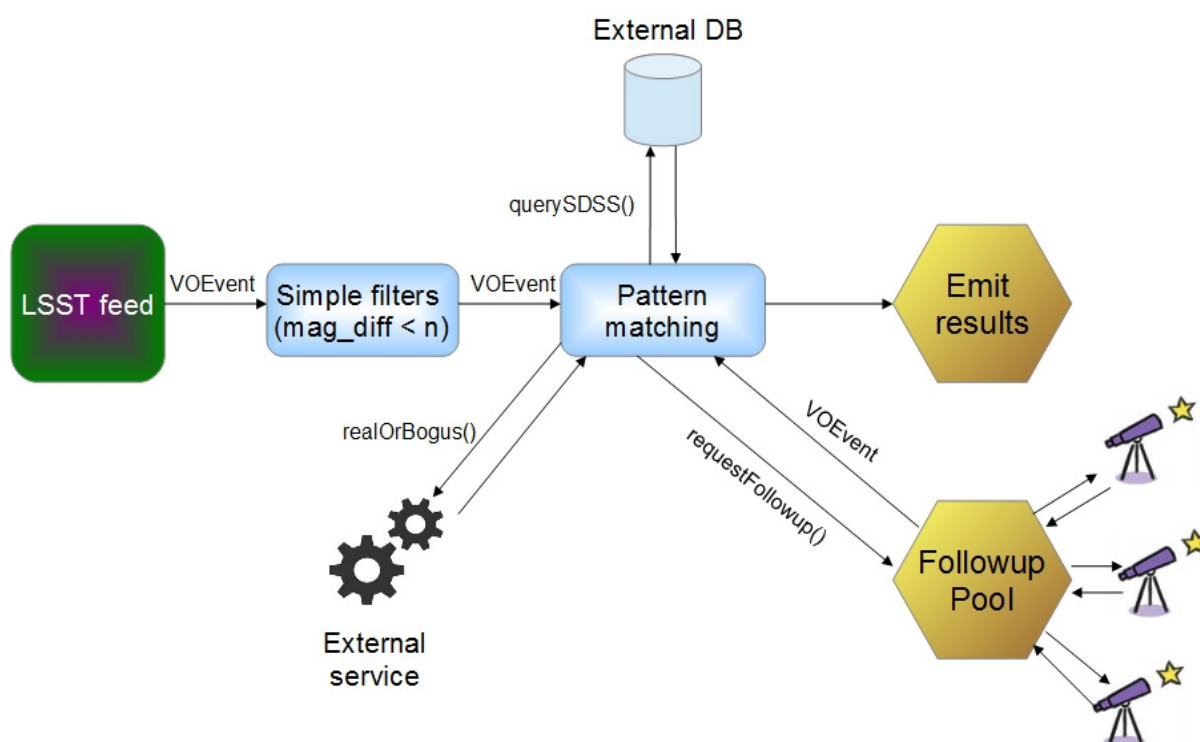
Kada agent nema dovoljno informacija na osnovu kojih bi doneo valjan zaključak, može zatražiti konsultaciju sa drugim aplikacijama, servisima, bazama podataka itd.

³ Apache Kafka je distribuirana platforma za obradu tokova podataka

Neki autori konceptualno razdvajaju "globalno stanje" unutar sistema od "eksternih servisa", kao kod Etzion & Niblett (2010). Kao primer globalnog stanja navodi se stepen uzbunjenosti na aerodromu. Agent za obradu događaja bi tako u zavisnosti od stepena uzbunjenosti donosio različite zaključke za identične ulazne događaje. Ipak, ako posmatramo distribuiranu arhitekturu kao celinu sastavljenu od tehnološki heterogenih delova, konsultacija sa bilo kojim modulom koji je osposobljen da učestvuje u razmeni i obogaćenju informacija o događaju se može tretirati na istom konceptualnom nivou.

2.3 Ilustracija arhitekture primenjene na astronomiju

Na slici 4 data je pojednostavljena ilustracija mreže za obradu događaja (engl. *event processing network*) na polju astronomije (Vujčić, 2014; Vujčić & Jevremović, 2020). Grafički model za gradivne blokove preuzet je iz knjige *Event Processing in Action* (Etzion & Niblett, 2010). Sa leve strane je proizvođač događaja, plavi zaobljeni pravougaonici predstavljaju agente (EPA), a žuti heksagoni kanale. Prvi agent primenjuje jednostavan filter na



Slika 4: Primer EPN arhitekture u astronomiji (Vujčić, 2014; Vujčić & Jevremović, 2020) iskazan pomoću gradivnih blokova definisanih u Etzion & Niblett (2010)

ulazni tok događaja, drugi agent obavlja detekciju paterna, konsultuje se po potrebi sa eksternim

servisima ili bazama, traži dodatna posmatranja (followup) i predaje rezultate kanalu koji ih emituje konzumentima.

2.4 Jezici za obradu tokova događaja

Jezici za obradu tokova događaja mogu se grubo podeliti na:

- jezike bazirane na relacionoj algebri (*sql-like*)
- jezike bazirane na pravilima (*if-then* tj. *when-then*)

Kao tipične predstavnike ova dva pristupa, predstavimo sintaksu jezika za procesiranje događaja u alatima Esper (Esper EPL) i Drools (Drools Fusion). Kao srodne jezike možemo još navesti SiddhiQL, Oracle Continuous QL, Apama EPL, EventScript itd.

2.4.1 Esper i Drools Fusion

2.4.1.1 Esper

Sintaksa Esper jezika za obradu događaja – EPL (EsperTech, n.d.) (Event Processing Language) – je SQL sintaksa sa proširenjima:

```
[annotations]  
[expression_declarations]  
[context context_name]  
[into table table_name]  
[insert into insert_into_def]  
select select_list  
from stream_def [as name] [, stream_def [as name]] [...]  
[where search_conditions]  
[group by grouping_expression_list]  
[having grouping_search_conditions]  
[output output_specification]  
[order by order_by_expression_list]  
[limit num_rows]
```

U okviru *from* klauzule može se definisati patern:

```
pattern [pattern_expression] [.view_spec] [.view_spec] [...]
```

Uvođenje konteksta radi se pomoću "okvira" (engl. *windows*):

```
win:length, win:length_batch, win:time, win:time_batch, win:time_length_batch,
win:time_accum, win:ext_timed, win:ext_timed_batch, ext:sort, ext:rank,
ext:time_order, std:unique, std:groupwin, std:lastevent, std:firstevent,
std:firstunique, win:firstlength, win:firsttime.
```

2.4.1.2 Drools Fusion

Drools Fusion je jezik baziran na pravilima (rule-based).

```
rule "rulename"
when
    // this pattern occurs
then
    // do something
end
```

Drools Fusion podržava vremenske operatore intervalne algebre

```
$eventA : EventA( this after[ 3m30s, 4m ] $eventB )
```

kao i kontekstualne "okvire":

```
EventFeed() over window:time( 2m )
```

2.4.1.3 Poređenje alata Esper i Drools

Oba alata su napisana u programskom jeziku Java i solidno su dokumentovani. Alati se lako integrišu u neko od vizuelnih okruženja za Javu (npr. Eclipse).

Slede zaključci nakon obavljenih testova iz 2013. pomoću Esper i Drools Fusion alata. Testovi su rađeni nad bazičnom konfiguracijom i jednostavnim primerima:

- Drools nema sintaksu za group by klauzulu, i zahteva zaobilazna rešenja što otežava agregaciju
- Esper formulacije (kao i SQL) su predstavljene kao tekstualne niske što otežava otklanjanje grešaka.
- Drools ima bolju podršku za temporalnu intervalnu algebru
- Drools je konceptualno jasniji i jednostavniji za početnike
- Esper podržava ulazne događaje u XML formi, ume da čita XSD šemu i primeni XPath upite

- Esper je izražajno moćniji, delom zbog fleksibilnosti samog SQL jezika, delom zbog velikog broja koncepata koji su ugrađeni u proširenja jezika
- Esper je znatno brži

2.4.1.4 Primer Esper sintakse na astronomskom scenariju

Primeri u ovom odeljku zasnovani su na radu Real-Time Stream Processing in Astronomy (Vujčić & Jevremović, 2020), objavljenom kao poglavlje u knjizi Knowledge Discovery in Big Data from Astronomy and Earth Observation (Skoda & Adam, 2020), i prilagođeni su potrebama ove disertacije.

Osnovni koncepti kompleksne obrade događaja prikazani su kroz astronomske upite u EPL-u. Primeri postepeno prelaze od jednostavnih izraza ka vremenskim okvirima, kontekstima i obrascima. Time se uvodi ideja deklarativnog opisa naučnog upita, koja se u kasnijim poglavljima razvija kroz semantički model i DSL sistema Alertissimo.

Osnove

Najpre definišemo osnovnu šemu toka podataka:

```
create schema LsstStream(diaObjectId int,
    diaSourceId int, ssObjectId int,
    filterName string);
```

U ovoj šemi diaObjectId označava jedinstveni identifikator nebeskog objekta. Polja diaSourceId i ssObjectId identifikuju pojedinačnu pojavu objekta van, odnosno unutar Sunčevog sistema i međusobno se isključuju, dok filterName označava optički filter korišćen pri posmatranju. Novootkriveni objekti još nemaju diaObjectId.

Projekcija se zadaje SELECT klauzulom, kao u SQL-u. U sledećem upitu broj ranije povezanih objekata označen je kao oldcount, a broj novih kao newcount:

```
select count(diaObjectId) as oldcount,
    sum(case
        when diaObjectId is null then 1
    end) as newcount,
    filterName
from LsstStream
```

WHERE klauzulom izdvajamo objekte koji ne pripadaju Sunčevom sistemu:

```
where diaSourceId is not null
```

Rezultati se zatim grupišu prema optičkom filteru:

```
group by filterName
```

Na grupu se može primeniti dodatni uslov, ovde kada newcount pređe zadati prag. Kao i u SQL standardu, alias iz SELECT klauzule ne može se koristiti u HAVING klauzuli:

```
having sum(case
    when diaObjectId is null then 1
end) > 100
```

Prozori ili okviri (Windows)

Vremenska dimenzija je inherentna obradi događaja, pa su podatkovni prozori ili okviri jedan od osnovnih CEP mehanizama. Oni ograničavaju skup događaja nad kojim se primenjuju pravila i omogućavaju zaključivanje unutar određenog vremenskog intervala. Njihova osnovna uloga ilustrovana je na slici 3.

Ako želimo da ograničimo izveštaj na vremenski okvir, možemo dodati:

```
from LsstStream.win:time(1 hour)
```

što daje broj objekata u kliznom vremenskom okviru u poslednjih sat vremena, osvežen sa svakim novim događajem.

Ako želimo da koristimo spoljne vremenske oznake umesto sistemskog vremena:

```
from LsstStream.win:ext_time(1 hour)
```

Statički vremenski okvir (nalik grupnoj (engl. *batch*) obradi) daje izveštaj jednom po satu:

```
from LsstStream.win:time_batch(1 hour)
```

Klizni okvir po dužini (broju događaja):

```
from LsstStream.win:length(1000)
```

Statički okvir po dužini:

```
from LsstStream.win:length_batch(1000)
```

Kontekst (Context)

Kontekst se koristi kada tok treba podeliti na nezavisne celine. Jednom definisan kontekst može se ponovo koristiti, čime se smanjuje ponavljanje koda.

```
create context RaDec partition by ra, dec
from LsstStream
```

Ovim izrazom događaji se dele prema položaju na nebu (rektascenziji i deklinaciji), tako da se događaji sa istim koordinatama obrađuju kao grupa koja pripada istom objektu.

Upotreba konteksta može zameniti `group by`:

```
context RaDec
select count(*) from LsstStream;
```

umesto:

```
select count(*) from LsstStream
group by ra, dec
```

Spajanje tokova (Joined Streams)

Za spajanje tokova koriste se INSERT INTO i JOIN konstrukcije. Kada tokovi sadrže različite tipove događaja, Esper nudi „variant stream“, kojim se oni izlažu kroz zajednički tok. To je tehnološki ograničen oblik ujednačavanja heterogenih ulaza; u Alertissimu se isti problem kasnije razmatra na nivou strukture i semantičke normalizacije podataka.

Primer definicije:

```
<variant-stream name="VariantStream">
  <variant-event-type name="LsstStream"/>
  <variant-event-type name="ZtfStream"/>
</variant-stream>
```

Ovo omogućava pojednostavljeno definisanje obrazaca:

```
pattern
[VariantStream->VariantStream]
```

čime se izbegava eksplicitno navođenje svih kombinacija tipova događaja.

Obrasci (Patterns)

Prepoznavanje obrazaca predstavlja jedno od osnovnih područja primene kompleksne obrade događaja i često obuhvata vremenske odnose između više događaja.

Primer je otkrivanje promene sjaja izražene promenom magnitude po satu:

```
dmdt = (m1-m2) / (t2-t1)
if (dmdt >= 1 and object was previously
    undetected)
then alert
```

Odgovarajuća transformacija toka u EPL-u glasi:

```
create schema LsstStream(ra double, decl double,
    flux double, diaObject boolean);
```

```

create variable double zeroPoint =
    -2.5*Math.log10(3631);
insert into LsstStreamTransformed
    select ra,
    decl,
    diaObject,
    -2.5*Math.log10(flux) - zeroPoint as mag,
    current_timestamp() as detectionTime
from LsstStream;

```

Na transformisanom toku zatim se primenjuje obrazac MATCH_RECOGNIZE:

```

select * from LsstStreamTransformed
match_recognize (
    partition by ra, decl

    measures A.ra as ra, A.decl as decl,
        A.detectionTime as firstDetection,
        B.detectionTime as secondDetection,
        A.mag as firstMag, B.mag as secondMag,
        Math.abs(B.mag - A.mag)/(
            (B.detectionTime-A.detectionTime)/3600000
        ) as ratePerHour

    pattern (A B)

    define
        A as A.diaObject=false,

        B as Math.abs(B.mag - A.mag)/(
            (B.detectionTime-A.detectionTime)/3600000
        ) >=1
)

```

Primer pokazuje kako se astronomski uslov može izraziti deklarativno, dok izvršni mehanizam preuzima njegovu primenu nad tokom događaja. U Alertissimu je ovo razdvajanje prošireno. DSL opisuje naučnu nameru, dok orkestracioni sloj bira servis i priprema fizički poziv; dobijeni odgovor zatim prevodi u zajedničku strukturu.

3 Orkestracija procesa u sistemima za obradu događaja

U ovom delu prikazaćemo teorijsku osnovu za spoj sistema baziranih na događajima i koncepata orkestracije procesa, kao polaznu tačku za razvoj srodnog sistema u astronomiji.

3.1 Evolucija sistema za orkestraciju i kontrolu tokova procesa/rada

Istorijski gledano, početni poslovni (engl. *enterprise*) sistemi oslanjali su se na čvrsto integrisane komponente i grupnu obradu, gde je tok upravljanja bio ugrađen u aplikativnu logiku. Kako su distribuirani servisi postajali sve brojniji i raznovrsniji, organizacije su počele da usvajaju radne tokove (engl. *workflow*) i procesne mehanizme za eksplicitno koordinisanje zadataka, koristeći notacije poput BPMN-a i konačnih automata za modelovanje poslovnih procesa i vođenje orkestracije (Singhal, 2024). Ovi mehanizmi su apstrahovali logiku poslovnog procesa od koda aplikacije, omogućavajući da procesi (poput zahteva i odobrenja, dodavanja korisnika, upravljanja inventarom itd) budu vizuelno modelovani, revidirani i prilagođeni bez potrebe za izmenom osnovnog sistema (Odofin et al., 2024). Može se reći da je ovo bio prvi veliki talas „orkestracije“ u poslovnim sistemima.

Sa eksplozijom razvoja sistema vezanih za veb i mikroservise, arhitektura zasnovana na događajima (EDA) pojavila se kao način za postizanje visokog obima obrade i odziva u realnom vremenu putem asinhronne razmene poruka i paterna objavljivanja i pretplate. Danas orkestracija predstavlja centralni nivo za koordinaciju heterogenih servisa, skladišta podataka i interakcija korisnika. Npr u finansijskim sistemima, mehanizmi orkestracije kombinuju tokove događaja koje kontroliše Kafka sa BPM mehanizmima za koordinaciju višefaznih radnih tokova i primenu regulatornih procesa poput KYC⁴, AML, i višestepenog odobravanja transakcija, uz obezbeđivanje mogućnosti revizije i usklađenosti sa propisima.

Nedavni radovi ističu orkestraciju uz podršku veštačke inteligencije i orkestraciju koja uzima u obzir kontekst (engl. *context-aware*), gde orkestratori prilagođavaju tokove na osnovu hijerarhije organizacije, konteksta u realnom vremenu i uvida dobijenih mašinskim učenjem (Talaseela, 2025). Ukupno gledano, istorijska putanja pokazuje da se orkestracija pomera od statičke, centralno definisane kontrole procesa ka dinamičkoj, inteligentnoj i adaptivnoj koordinaciji gde događaji ne

4 KYC (know your customer) i AML (anti-money laundering) su finansijske regulatorne procedure usmerene protiv pranja novca i finansijskog kriminala i terorizma.

predstavljaju samo ulaz u predefinisani sistem već mogu promeniti i stanje sistema i uticati na kontekst.

Tabela 3: Istorijska evolucija orkestracije kroz različite softverske arhitekture

Vrste softverskih arhitektura	Uloga orkestracije
Klasični sistemi za orkestraciju/workflow	Vizuelno modelovanje poslovnih procesa, centralizovana koordinacija
Sistemi orijentisani na događaje	Koordinacija bazirana na događajima, veliki obimi podataka, orkestracija-usklađivanje-revizija u realnom vremenu
Serverless, cloud	Tokovi procesa zasnovani na DSL jezicima, tokovi koji se mogu ponovo koristiti
AI, context-aware	Adaptivna orkestracija, procesi kontrolisani metodama mašinskog učenja, korisnička interakcija putem NLP/LLM modula

3.2 Uloga orkestratora

Koncepti arhitekture sistema baziranih na događajima komplementarni su sa orkestracijom i kontrolom tokova procesa/rada (engl. *workflow*). Mehanizmi za definisanje, koordinaciju i kontrolu funkcionišu kao "nad-sloj" u scenarijima koji zahtevaju više koraka u ekosistemu koji se sastoji od više heterogenih aktera. Orkestrator je zadužen za integraciju servisa i mikroservisa u koherentan radni proces (Ahmed et al., 2021). To podrazumeva proveru dostupnosti (engl. *availability*) i sposobnosti (engl. *capability*) određenog servisa određenog aktera ovakvog distribuiranog sistema. Takođe, orkestrator objedinjuje semantički prostor domena, zadatak koji je često kompleksan zbog neusklađenosti ulaznih i izlaznih funkcija komponenti i nedostatka standardizovanog rečnika podataka.

Pregled uloga orkestratora predstavice u tabeli 4

Tabela 4: Uloge orkestratora u sistemima za obradu događaja

Uloga	Primeri u implementaciji
<i>Definisanje tokova procesa/rada</i>	BPMN ⁵ , konačni automati, domenski-specifični jezici
<i>Reakcije na događaje</i>	Pravila, ECA ⁶ trigeri,
<i>Koordinacija servisa</i>	sekvencionisanje/paralelizacija, (pre)usmeravanje mikroservisa
<i>Upravljanje stanjima / Održavanje konzistencije</i>	Identifikatori korelacije, Saga, praćenje događaja (engl. <i>event sourcing</i>)
<i>Upravljanje greškama</i>	Ponovni pokušaji, vremenska ograničenja, redovi za neuspešno obrađene poruke

3.3 Evaluacija sinergije sistema za obradu događaja i orkestracije procesa

Može se zaključiti da uključivanje komponenti za orkestraciju u sistem za obradu događaja povećava efikasnost tokova procesa/rada. Zajedno ovakvi sistemi nude i druge značajne prednosti nad tradicionalnim aplikacijama, poput adaptabilnosti, otpornosti, dinamičkih (promena) pravila, pametnog prilagođavanja pravila itd. Mnogi izvori iz literature navode konkretne metrike gde su znatno poboljšane karakteristike sistema. Izdvajamo neke od primera iz literature:

Izvršenje procesa

- Integracija orkestracije u sisteme za obradu događaja skraćuje prosečno vreme izvršenja celovitih procesa za 47% i redukuje potrebu za ljudskom intervencijom za 58% (Talaseela, 2025).
- U aplikacijama za elektronsku trgovinu, redukovano je ukupno vreme obrade narudžbine za 72% i povećana preciznost sinhronizacije podataka na 99.96% (Talaseela, 2025).

⁵ BPMN (Business Process Model and Notation) – grafička reprezentacija specifikacije poslovnih procesa

⁶ ECA (event-condition-action) – aktivna pravila u sistemima za obradu događaja

Latencija i odgovor u realnom vremenu

- Kompleksni poslovni sistemi bazirani na događajima mogu da obrade preko 42,000 događaja u minuti sa latencijom ispod 15ms (Talaseela, 2025) i vreme reakcije u realnom vremenu od ~1.2s (Gothi, 2025) (od događaja do akcije), za red veličine kraće nego u tradicionalnim sistemima.
- Analitički, monitoring sistemi, i slični pomoću arhitekture za obradu događaja mogu postići vremena obrade ispod milisekunde i obrađivati milione događaja u sekundi, i pritom očuvati otpornost sistema (Kundavaram, 2024).

4 Astronomija vremenskog domena

Astronomija vremenskog domena (engl. *time domain astronomy*) bavi se proučavanjem astronomskih fenomena kod kojih je vreme fundamentalna komponenta (Zwicky, 1933). U najširem smislu njen predmet je otkrivanje promenljivih objekata na nebu na osnovu nekoliko sukcesivnih posmatranja, u čemu su digitalne pretrage neba donele ključna kvalitativna i kvantitativna poboljšanja (Djorgovski et al., 2001; Eyer & Cuypers, 2000). Broj i vrsta posmatranja zavise od tehnoloških kapaciteta i naučnih metoda koji su dostupni da bi se doneo valjan zaključak o svrstavanju objekta koji je meta posmatranja.

4.1 Promenljivi astronomski objekti

Promenljivost objekata na nebu uočena je u starim civilizacijama, i uglavnom se odnosila na promenu pozicije tj. kretanje nebeskih tela bliskih Zemlji (North, 2008). Plinije u I veku daje naziv *nova* za eksploziju zvezde (Pliny, 1855). U XI veku japanski, kineski i arapski astronomi primetili su pojavu *supernove*, tj. eksplozije zvezde koja izbacuje ogromnu energiju za izuzetno kratak vremenski period (Stephenson & Green, 2002). Na ovakve događaje je trebalo čekati vekovima da bi se poklopile okolnosti pod kojima se mogu uočiti golim okom. Tycho Brahe je u drugoj polovini XVI veka uočio razliku u paralaksi⁷ upoređujući supernovu i komete (za SN je paralaksa znatno manja) i time dao dokaze da univerzum izvan Sunčevog sistema nije statičan (Brahe, 1573; Thoren & Wilson, 1991). I posle pojave prvih teleskopa, šanse da se ovakvi objekti uoče su ostale male zbog uskog ugla posmatranja.

Svaki je astronomski objekat kontinuirano promenljiv - po kretanju, fizičkoj transformaciji, promeni sjaja itd. Ipak, praktična podela zavisi od detektabilnosti određene vrste promenljivosti i naučnog cilja konkretnog posmatranja ili projekta. Svetlosne promene astronomskih objekata prate se putem sukcesivnih merenja koja zajedno formiraju *krivu sjaja*, "najvažniji pojedinačni grafikon u proučavanju promenljivih zvezda" (Mattei & Percy, 2001), preko koje se mogu otkriti paterni periodičnosti ili uočiti nedostatak periodičnosti.

Kepler projekat (Borucki, 2016) je, na primer, usko specijalizovan za tranzite malih planeta i promene krivih sjaja u okviru Mlečnog puta.

⁷ Paralaksa je razlika u poziciji zavisna od ugla gledanja

Na slici 5 je prikazan jedan vid podele promenljivih objekata. Ovakva klasifikacija uključuje i objekte sunčevog sistema, tj asteroide. Ovde je zgodno napomenuti da se iz naučnih i praktičnih razloga, o kojima će kasnije biti više detalja, objekti sunčevog sistema tretiraju i obrađuju na poseban način.

Savremeni sveobuhvatni ("sinoptički") projekti, poput LSST-a dele promenljive objekte u nekoliko podkategorija (LSST Science Collaboration, 2009):

- Tranzijentni objekti
- Objekti čiji se sjaj menja zbog efekta gravitacionog sočiva
- Promenljive zvezde
- Planete u tranzitu

4.1.1 Tranzijentni objekti

tran·sient

/ˈtrənSHənt, ˈtranzēənt/

adjective

1. lasting only for a short time; impermanent.

Tranzijentni (kratkoročni, prolazni) astronomski događaj se definiše kao astronomski fenomen čije se trajanje može uočiti ljudskom percepcijom, najčešće na skali od sekundi do nekoliko godina (Kulkarni, 2012).

Nebeski objekat nad kojim je detektovan (kompleksni) tranzijentni događaj nazivamo tranzijentnim objektom, ili tranzijentom.

Postoji jasna terminološka razlika između tranzijentnog objekta i objekta koji je "samo" promenljiv: tranzijent u fizičkom smislu menja svoj karakter tokom samog događaja i njegova fizička svojstva su ireverzibilna; promenljivi objekat koji nije tranzijent menja sjaj iz nekog od niza mogućih razloga ali ne i fizički karakter u bitnoj meri (Paczynski, 2000).

Tranzijentni događaji su nasilni događaji u svemiru, poput kolizija i eksplozija. Naučni interes za kataklizmičke tranzijente prevazilazi samu fascinaciju događajem i potrebu za njegovom karakterizacijom. Pokazano je npr. da supernove određenog tipa zahvaljujući svojim uniformnim karakteristikama mogu da služe kao indikatori distance u svemiru ("standardne sveće"⁸) kao i da mogu da budu ključ za razumevanje drugih ključnih astrofizičkih fenomena poput tamne energije⁹, širenja svemira itd (Perlmutter et al., 1999; Schmidt et al., 1998).

Od savremenih projekata se očekuje da omoguće detekciju tranzijenata koji ne potpadaju pod trenutnu klasifikacionu nomenklaturu (Ivezić et al., 2019).

⁸ Standardne sveće su astronomski objekti koji pripadaju klasama objekata sa poznatom apsolutnom luminoznošću, kao što su određeni tipovi supernova i promenljivih zvezda. Pomoću njih se određuje distanca, kao funkcija izmerene magnitudo m i apsolutne magnitudo M .

⁹ Tamna energija je hipotetička sila koja podstiče ubrzano širenje svemira

4.1.2 Događaji izazvani efektom gravitacionog sočiva

Efekat gravitacionog sočiva predstavlja optičko izobličenje koje se pojavljuje za vreme preklapanja astronomskih objekata na različitim distancama (Schneider, 2005). Najčešće, izvor je objekat na velikoj udaljenosti dok kao sočivo služi objekat na srednjoj udaljenosti. Do zakrivljenja putanje svetlosti dolazi usled jakog uticaja gravitacionih polja.

Iako se efekat gravitacionog sočiva teško uočava, posmatranja ovakvih fenomena su postala moguća uz automatizovano praćenje velikog broja objekata.

Jako gravitaciono sočivo projektuje višestruke i/ili deformisane slike udaljenog objekta (kvazara, galaksije itd) (Kochanek, 2006).

Slabo gravitaciono sočivo blago deformiše objekte koji se nalaze u polju iza sočiva (Hoekstra et al., 2002). Ukoliko se detekcija vrši pomoću prostorne analize, koja se može primeniti na malim udaljenostima, tehnika detekcije se naziva *mesolensing* (Di Stefano, 2008). Kada su objekti udaljeniji mora se pristupiti analizi u funkciji vremena i promene položaja objekata koja se naziva mikrolensing (engl. *microlensing*; Paczynski, 1986). Ova tehnika je od posebnog interesa za astronomiju vremenskog domena.

Naučni ciljevi pri istraživanju događaja izazvanih efektom gravitacionog sočiva uključuju proveru hipoteza o tamnoj materiji¹⁰ pomoću MACHO¹¹ objekata, detekciju udaljenih planeta, istraživanja populacija zvezda itd (Tisserand et al., 2007).

Tip sočiva	Prošlost po deceniji po deg ²	Sadašnjost po deceniji po deg ²	Budućnost po deceniji po deg ²	Budućnost po deceniji za 150 deg ²
M patuljci	2.2	46	920	1.4×10^5
patuljci	0.051	1.1	22	3200
T patuljci	0.36	7.6	150	2.3×10^4

¹⁰ Tamna materija je hipotetička vrsta materije koja utiče na fenomene vezane za izmenjeno ponašanje sile gravitacije u udaljenom svemiru

¹¹ MACHO (MAssive Compact Halo Object) je objekat koji učestvuje kao sočivo u microlensing-u.

WD	0.4	8.6	170	2.6×10^4
NS	0.3	6.1	122	1.8×10^4
BH	0.018	0.38	7.7	1200

Tabela 5: Podaci i predviđanja o objektima detektovanim tehnikom mikrolensinga (Di Stefano et al., 2008).

4.1.3 Promenljive zvezde

Zvezde sa promenljivim sjajem mogu biti intrinističkog i ekstrinističkog tipa, u zavisnosti od toga da li se promena sjaja dešava zbog prirode same zvezde ili njene okoline (Eyer & Mowlavi, 2007).

Ekstrinistički tip promenljivih zvezda odaje utisak promene sjaja usled orbitiranja (povremenog pomračenja, npr. kod binarnih zvezda) ili fenomena vezanih za rotaciju (Prša et al., 2011).

Kod intrinističkog tipa, širenje i skupljanje spoljnih slojeva zvezde prouzrokuje pulsiranje svetlosti. Ovakvo ponašanje se razlikuje po periodama pulsiranja i oblicima krivih sjaja. Način pulsiranja zvezde zavisi od njenog evolutivnog doba (Catelan & Smith, 2015). I kod stabilnih zvezda kao što je Sunce, koje se nalaze u stanju hidrostatičkog ekvilibrijuma¹² dolazi do neznatnog pulsiranja sa dugačkim periodama (Aerts et al., 2010). Od specifičnog interesa za većinu naučnih programa su intrinističke promenljive zvezde sa značajnim amplitudama promene sjaja i relativno kratkim periodama.

U tabeli 6 vide se klase pulsirajućih zvezda u odnosu na periode i amplitude sjaja.

4.1.4 Planete u tranzitu

U savremenim projektima, detekcija planeta vrši se u velikoj većini fotometrijom u tranzitu (Winn, 2010). Pošto su planete vrlo blede objekti, metode direktnog posmatranja, ugaone brzine, mikrolensing-a itd daju slabije rezultate.

Velike planete provode 1-5% vremena orbitiranja u tranzitu i tada izazivaju parcijalno pomračenje

¹² Hidrostatički ekvilibrijum je balans između gravitacione sile zvezde koja privlači zvezdani materijal ka jezgru i pritiska radijacije i gasa

svoje matične zvezde tj promenu krive sjaja (Haswell, 2010).

Vrsta promjenljive zvezde	Perioda	Amplituda sjaja (mag)
α Cyg	1 - 50 d	nekoliko 0.1
β Cephei	2 - 12 h	do 0.05
Cepheids	2 - 70 d	0.1-1.5
δ Scuti	30 min - 6 h	do nekoliko 0.1
EC 14026	80 s - 8 min	do 0.03
γ Dor	8 h - 3 d	do nekoliko 0.01
GW Vir	5 - 80 min	do 0.2
PG1716 "Betsy"	30 - 150 min	do ~ 0.01
roAp	5 - 20 min	do 0.01
RR Lyrae	5 h - 1.1 d	0.2-2
RV Tau	30 - 150 d	1-3
SPB	0.5 - 5 d	do 0.03
SR, Mira	50 - 1000 d	do 8
V777 Her	2 – 16 min	do 0.2
W Vir	1 - 20 d	0.3-1.2
ZZ Ceti	0.5 – 25 min	do 0.2

Tabela 6: Najčešće pulsirajuće zvezde sa periodama (s, min, h, d - sekunda, minut, sat i dan respektivno) i razlikama u sjaju (u magnitudama) (Eyer & Mowlavi, 2007).

5 Klasifikacija astronomskih objekata u *BigData* eri

Jedan od glavnih ciljeva analize astronomskih podataka je klasifikacija astronomskih objekata tj. svrstavanje određenog objekta na osnovu njegovih karakteristika u klasu koja okuplja srodne objekte.

Klasifikacija se može bazirati na već formiranom teoretskom/empirijskom modelu ili na obzervacionom principu: podaci govore sami za sebe i dovoljna su mera istinitosti klasifikacije bez obzira na naše (ne)razumevanje modela.

U praksi se ova dva pristupa najčešće kombinuju – npr. teoretski model se usavršava statističkim zaključcima. Savremeni trendovi dodatno brišu granicu između ova dva pristupa, kroz upotrebu hibridnih modela koji kombinuju postojeće modele i duboko učenje, npr. PIML (*physics-informed machine learning*) (Smith & Geach, 2023; Meskhidze, 2024).

Naučni interes za klasifikaciju astronomskih objekata može se sumirati kroz četiri tačke (Bloom & Richards, 2011), uz napomenu da "ne postoje dva identična objekta u prirodi i da su granice između klasa inherentno nejasne":

- Razumevanje fizičkih procesa kroz razlike u varijabilnosti, uključujući unutrašnju strukturu zvezda.
- Korist od određenih klasa koja nije u direktnoj sprezi sa samom prirodom varijabilnosti, kao što je merenje kosmičkih distanci pomoću određenih klasa objekata.
- Demografija velikog broja objekata može doneti bolje razumevanje njihovog životnog ciklusa.
- Otkrivanje rariteta i anomalija* unutar postojećih klasa, odnosno formulisanje novih klasa.

* U poslednjim godinama, detekcija anomalija postaje centralni cilj, posebno u kontekstu otkrivanja novih tipova tranzijenata koje prethodni modeli ne mogu da obuhvate (Villar et al., 2021; Muthukrishna et al., 2022).

Barijeru više ne predstavlja dostupnost podataka, već sposobnost da se iz raspoloživih podataka

ekstrahuju korisne i pouzdane informacije. Ovakav pristup se naziva i "četvrta paradigma"(Hey et al., 2009):

- 1) pre 1000 godina: eksperimentalna nauka – opis prirodnih fenomena
- 2) nekoliko vekova unazad: teorijska nauka – fizički zakoni
- 3) polovina XX veka: numeričke simulacije
- 4) danas: otkrivanje pravilnosti (data-intensive science, data-driven discovery):
 - od "nemaštine" do preopterećenosti podacima
 - od centralizovanih do heterogenih izvora
 - od setova podataka do tokova podataka
 - od povremene do "real-time" analitike
 - od vlasništva nad podacima do vlasništva nad ekspertizom

Ova paradigma je posebno izražena u velikim astronomskim projektima poput ZTF-a i LSST-a tj. njihovim ekosistemima, gde je jedan od glavnih naučnih ciljeva da se u realnom vremenu identifikuju zanimljivi promenljivi objekti u moru "običnih" objekata.

Izazovi analize astronomskih podataka u BigData eri su (Ivezic, 2015):

- veliki obim podataka (desetine petabajta numeričkih podataka)
- veliki broj objekata (desetine milijardi)
- višedimenzionalni prostor (desetine hiljada atributa)
- nepoznate statističke distribucije
- vremenske serije
- heteroskedastični, okrnjeni i nedostajući podaci

- nepouzdana podaci zbog slučajnih ili sistematskih grešaka

Savremenoj astronomskoj nauci izazov ne predstavljaju samo metode klasifikacije koje bi operisale nad kvalitetnim podacima. Problem je i kako ih prilagoditi da dobro rade nad loše semplovanim podacima sa potencijalno niskim odnosom signala i šuma, što je posebno izraženo u ranoj fazi detekcije tranzijenata, gde su dostupne samo parcijalne svetlosne krive i odluke moraju biti donete uz visok stepen neizvesnosti (Ivezić et al., 2014; Muthukrishna et al., 2019; Shah et al., 2025). Takođe, savremeni izazovi uključuju i promene statističkih svojstava podataka i odnosa koje model opisuje tokom vremena (engl. *concept drift*) i potrebu za kontinuiranim doterivanjem modela u realnom vremenu – ovo će biti tipično za LSST i okruženje (Förster et al., 2021; Cabrera-Vives et al., 2024).

5.1 Karakter astronomskih podataka

Kvalitet podataka iz vremenskih serija zavisi od kadenca i preciznosti snimanja. Da bi se dobio kvalitetan uzorak, kadenca snimanja bi trebalo da bude optimizovana za naučni program praćenja tranzijenata. Pošto se tranzijenti razlikuju po amplitudi magnitude (sjaja) i periodi, za kvalitetne podatke potrebno je dodatno prilagođavanje parametara posmatranja konkretnom naučnom cilju. To još uvek nije dovoljno jer često treba ograničiti deo neba koji se posmatra da bi teleskop stigao da ispuni potrebnu kadencu (npr. Kepler: planete u tranzitu oko 150K zvezda). Instrument mora posedovati karakteristike adekvatne potrebama naučnog programa (ugao gledanja, osetljivost na svetlo, dužina ekspozicije), a one su često u međusobnoj koliziji (kratka ekspozicija ili širok ugao -> ne vide se blede objekti; duga ekspozicija ili uzak ugao -> duga kadenca; velika osetljivost na svetlo -> nedovoljna oštrina, preteran šum; itd.).

Veliki savremeni astronomski projekti koncipirani su, iz naučnih, ekonomskih i političkih razloga, kao "sveobuhvatni" tj pokušavaju da zadovolje više naučnih programa u isto vreme (npr. LSST: spoljni pojas sunčevog sistema, "statičko" nebo, tranzijenti, asteroidi itd.) (LSST Science Collaboration, 2009). Takvi projekti su potencijalno veoma korisni za proučavanje tranzijenata zbog količine podataka koje će dopremiti i sposobnosti da zabeleže blede objekte u širokom vidnom polju. U isto vreme, takvi projekti neće isporučivati savršene podatke o tranzijentima naučnoj zajednici, već se od same zajednice traži istraživačka domišljatost da koncipira alate koji će umeti da prave klasifikaciju koristeći više heterogenih izvora podataka. LSST dodatno uvodi paradigmu *alert-driven astronomy*, gde se analiza pokreće događajem umesto unapred definisanim

eksperimentom (Förster et al., 2021).

Sama priroda fotometrijskih posmatranja često nije dovoljna za klasifikaciju već su potrebni i spektroskopski i drugi podaci, koje moraju isporučiti drugi teleskopi. U praksi, ovo dovodi do razvoja kompleksnih mreža teleskopa i automatizovanih sistema za zakazivanje posmatranja, tzv. TOM sistema (Narayan et al., 2018). Od posebnog je značaja naglasiti ovu potrebu za dodatnim (spektroskopskim) posmatranjima. Alati za svrstavanje astronomskih objekata moraju operisati u (skoro) realnom vremenu da bi se zahtev za dodatno posmatranje na fizički udaljenom instrumentu mogao ispuniti, tj. da bi taj instrument mogao da stigne da isprati fenomen koji se upravo odvija i koji često ima kritično kratko vreme trajanja (Shah et al., 2025; Möller et al., 2024). Ovakva dodatna posmatranja odnose se na tranzijentne objekte (poglavlje 4.1.1), koji bespovratno menjaju svoje fizičke karakteristike.

5.2 Klasifikacija pomoću mašinskog učenja

Klasifikacija pomoću metoda mašinskog učenja i otkrivanja pravilnosti u podacima (engl. *data mining*) po pravilu ne uzima postojeći teoretski model za polazno stanovište već operiše nad *n*-dimenzionalnim vektorima svojstava (engl. *feature*) (Bishop, 2006). Kao što je opštepoznato, mašinsko učenje može biti:

- **supervizovano**, gde se na osnovu prethodno označenih primeraka klasa induktivno trenira mehanizam za zaključivanje, koji kasnije vrši dedukciju nad neoznačenim podacima
- **nesupervizovano**, poznato i kao klasterovanje koje ne koristi unapred definisane klase već pokušava da detektuje nepoznate pravilnosti u prostoru svojstava

U supervizovanim metodama cilj je maksimizacija prediktivne moći klasifikacije novih izvora: performanse različitih supervizovanih metoda mogu se evaluirati i vizualizovati pomoću ROC krivih (Fawcett, 2006), dok se u novije vreme više pažnje posvećuje kalibraciji verovatnoća (Zadrozny & Elkan, 2002) i ranom prepoznavanju događaja u realnom vremenu (Muthukrishna et al., 2019). Mnoge klasične metode se koriste u astronomiji (Bayes, Naive Bayes, KDE, GM, QDA, LDA, SVM, KNN, ANN, stabla odluke, EM itd) (Bloom & Richards, 2011; Dubath et al., 2011; Kim & Bailer-Jones, 2016). Međutim, u LSST/ZTF eri, ovi pristupi se dopunjuju ansambl metodama i dubokim učenjem, što je posebno važno za rad sa heterogenim i nepotpunim podacima

(Förster et al., 2021; Bassi et al., 2021).

5.3 Savremene metode klasifikacije za LSST/ZTF

5.3.1 Ansambl metode

Random Forest (RF) i savremeni "gradient-boosting" algoritmi (npr. XGBoost, LightGBM) služe kao jaki polazni modeli za kataloge sa tabličnim atributima, poput ZTF/LSST kataloga. Oni dobro podnose heteroskedastične, okrnjene i delimično nedostajuće podatke, što je tipično za astronomske podatke (Singhal, 2024; Odofin et al., 2024).

U mnogim radovima još iz 2010-ih pokazano je da RF nadmašuje klasične metode (SVM, ANN) na istim skupovima podataka. Debosscher et al. (2007) predstavljaju metode (Bayesian Network, Gaussian Mixture, SVM, ANN) za klasifikaciju više klasa promenljivih zvezda nad ulaznim podacima iz heterogenih izvora. Koristeći iste podatke, Richards et al. (2011) pomoću drugačijih metoda ostvaruju smanjenje broja pogrešnih klasifikacija za 24%. Od klasičnih metoda, najviše uspeha imala je RF metoda u kombinovanom prostoru periodičnih i neperiodičnih svojstava, dok su sve metode bolje od najbolje metode korišćene u radu Debosscher et al. (2007). Sledeći primer je UPSILoN klasifikator (Kim & Bailer-Jones, 2016) koji koristi RF nad 16 atributa svetlosnih krivih za klasifikaciju periodičnih promenljivih zvezda iz različitih kataloga (OGLE, MACHO, LINEAR i ASAS), gde je postignuta visoka preciznost. Recentan primer je ALerCE broker koji koristi Balanced Hierarchical Random Forest (BHRF) treniran na 429 ručno izvedenih atributa iz ZTF svetlosnih krivih za klasifikaciju u 15 klasa (promenljive zvezde, tranzijenti, AGN i dr.) u realnom vremenu (Förster et al., 2021).

5.3.2 Sekvencijalno duboko učenje

Za svetlosne krive iz ZTF i budućeg LSST-a, koje su često kratke, neredovne i šumovite, koriste se rekurentne neuronske mreže (RNN, LSTM, GRU) i 1D konvolutivne mreže. One eksplicitno modeluju vremensku strukturu i mogu da rade direktno na nizovima posmatranja, umesto na ručno definisanim statističkim atributima. To je ključno u uslovima rane detekcije kada su dostupne samo parcijalne serije i odnos signal/šum je nizak (Muthukrishna et al., 2019; Shah et al., 2025; Charnock & Moss, 2017). Konkretni primer je RAPID (Real-time Automated Photometric Identification (Muthukrishna et al., 2019)) - klasifikator zasnovan na dubokoj RNN arhitekturi sa GRU

jedinicama (za pamćenje stanja) koji klasifikuje 12 tipova tranzijenata već od prvog dana nakon alerta, bez potrebe za ručnom ekstrakcijom atributa, i koji funkcioniše nad ZTF tokom podataka u realnom vremenu. Analogno, Charnock & Moss (2017) su pokazali efikasnost dubokih RNN-ova za klasifikaciju supernova.

Hibridne arhitekture (1D CNN + RNN) kombinuju lokalno detektovanje paterna (bljeskovi, kratki tranzijenti) sa dugoročnim kontekstom (postepene promene), što ih čini pogodnim za klasifikaciju supernova, AGN varijabilnosti i drugih LSST/ZTF tranzijenata. Primer su klasifikatori razvijeni u okviru Fink brokera, koji kombinuju SuperNNova (RNN-zasnovani klasifikator supernova) i CATS (klasifikator zasnovan na dubokom učenju za LSST podatke) za klasifikaciju opsežne taksonomije tranzijenata u realnom vremenu (Möller et al., 2024; Bassi et al., 2021).

5.3.3 Transformer modeli za vremenske serije

Među najnovijim trendovima je primena transformer multimodalnih arhitektura na neredovne i višekanalne svetlosne krive (više filtera, više instrumenata) i dugačke sekvence vremenskih serija. Paradigmatičan primer je ATAT (Astronomical Transformer for time series And Tabular data (Cabrera-Vives et al., 2024)), razvijen od strane ALeRCE brokera za ELAsTiCC¹³ izazov koji simulira LSST tok podataka. ATAT koristi dva transformer modela - jedan za svetlosne krive i jedan za tabelarne podatke, sa *macro F1* pokazateljem od 82.9 preko 20 klasa, nadmašujući BHRF model (79.4) treniran na 429 atributa, uz 400 puta kraće vreme zaključivanja (Cabrera-Vives et al., 2024). Takođe, rad Gupta et al. (2025) pokazuje da "transfer learning", metod unutar transformer modela smanjuje potrebu za označavanjem ZTF podataka za 95% uz zadržavanje iste tačnosti (Gupta et al., 2025).

5.3.4 Anomalije i otkrivanje retkih tipova događaja

Detekcija anomalija ima centralnu ulogu u ZTF/LSST eri, gde se od klasifikacionih sistema očekuje da prepoznaju i „nepoznate nepoznate“ tipove tranzijenata (Villar et al., 2021; Muthukrishna et al., 2022).

Pored klasičnog klasterovanja, koriste se:

- autoenkoderi i varijacioni autoenkoderi za učenje kompresovanih reprezentacija „uobičajenog“ ponašanja - npr. Villar et al. (2021) koriste varijacioni rekurentni autoenkoder

13 ELAsTiCC (Extended LSST Astronomical Time-series Classification Challenge) je set simuliranih LSST podataka namenjen razvoju i obučavanju brokera. Podaci su emitovani u tokovima.

treniran na samo 1% PLAsTiCC¹⁴ podataka za enkodovanje svetlosnih krivih Rubin/LSST tranzijenata, a zatim Isolation Forest za rangiranje anomalija; algoritam uspešno identifikuje veći indeks anomalije za ređe klase (Villar et al., 2021)

- Isolation Forest i slični detektori odstupajućih vrednosti kao brze, skalabilne metode za petabajtne kataloške podatke - primer je i AHA (Anomaly Hunter for Alerts) (Iskandarli et al., 2026), algoritam koji primenjuje tri nezavisna autoenkodera (na atributima objekata, isečcima slika i svetlosnim krivama) - trenirana na "normalnim" objektima - na ZTF alert stream putem Lasair brokera, identifikujući ~1–5 neobičnih kandidata po noći.

ALeRCE detektor anomalija (Perez-Carrasco et al., 2023) koristi "penultimate layer" neuronske mreže klasifikatora kao latentni prostor za detekciju anomalija, uz predloženu metodu Multi-Class Isolation Forests (MCIF) koja trenira zasebne Isolation Forest modele za svaku klasu. Ove metode dopunjuju supervizovane klasifikatore time što izdvajaju objekte koji ne pripadaju nijednoj poznatoj klasi.

5.3.5 Multi-modalno učenje

ZTF i LSST generišu astronomske notifikacije koje se moraju kombinovati sa dodatnim kataloškim, spektroskopskim i drugim podacima. Savremeni modeli zato sve češće koriste multimodalno učenje. Primer je AppleCiDER (Applying Multimodal Learning to Classify Transient Detections Early) (Junell et al., 2025), metod koji integriše četiri modaliteta - fotometriju, isečke slika, metapodatke i spektre - koristeći dva Transformer enkodera za fotometriju, multimodalni CNN za slike i metapodatke, i CNN za spektre (Junell et al., 2025). Takođe, Maven (A multimodal foundation model for supernova science) (Zhang et al., 2024) je multimodalni klasifikator supernova koji usaglašava fotometrijske i spektroskopske podatke.

¹⁴ PLAsTiCC (Photometric LSST Astronomical Time-series Classification Challenge) je set simuliranih LSST podataka razvijen za potrebe razvoja i treninga ML algoritama. Za razliku od ELAsTiCC, podaci su statični.

6 Ekosistem aplikacija vezanih za LSST/Rubin astronomski projekat

U ovom poglavlju daćemo prikaz ekosistema aplikacija vezanih za astronomske događaje, a uže vezanih za LSST/Rubin projekat. Veći deo ovog poglavlja predstavlja prevedenu i prilagođenu verziju rada "*An overview of astronomical transient brokers in Rubin era*" (Vujčić et al., 2025).

Opservatorija Vera C. Rubin, ranije poznata kao *Large Synoptic Survey Telescope* (LSST), približava se operativnoj fazi, a svi akteri u naučnom i softverskom ekosistemu koji je okružuje nalaze se u završnim fazama priprema. Jedan od tipova podataka koje će Rubin javno isporučivati jesu obaveštenja u realnom vremenu (engl. *alerts*) - tok podataka o tranzijentnim astronomskim događajima. Ova "upozorenja" će biti distribuirana javnosti putem sedam odobrenih softverskih aplikacija koje preuzimaju tok podataka i pružaju dodatne funkcionalnosti.

6.1 Uloga brokera u astronomiji vremenskog domena

Opservatorija Vera C. Rubin -Legacy Survey of Space and Time, ranije poznata kao Large Synoptic Survey Telescope (LSST (Ivezić et al., 2019)) – približava se operativnoj fazi. Notifikacije o događajima u realnom vremenu, odnosno tok podataka o tranzijentnim astronomskim pojavama, predstavljaju jedan od dva glavna tipa podataka koje će LSST isporučivati (Juric et al., 2015).

Tok podataka će biti javno dostupan, ali zbog infrastrukturnih razloga i ograničenja protoka LSST je odobrio sedam posredničkih (engl. *community broker*, u daljem tekstu broker) sistema - softverskih platformi koje preuzimaju, transformišu, obogaćuju i redistribuiraju tok - da deluju kao posrednički sloj između izvornog toka podataka i naučne zajednice. U najširem smislu, zadatak brokera nije da bude konačni, odlučujući naučni arbitar, već da deluje kao posrednik, smanjujući tok na manji broj događaja sa najvećim potencijalom za određeni naučni slučaj i nudeći mogućnosti automatizacije za programsko povezivanje sa drugim alatima u procesu. Imajući to u vidu, brokeri se razlikuju u svojim funkcionalnostima i u arhitekturi.

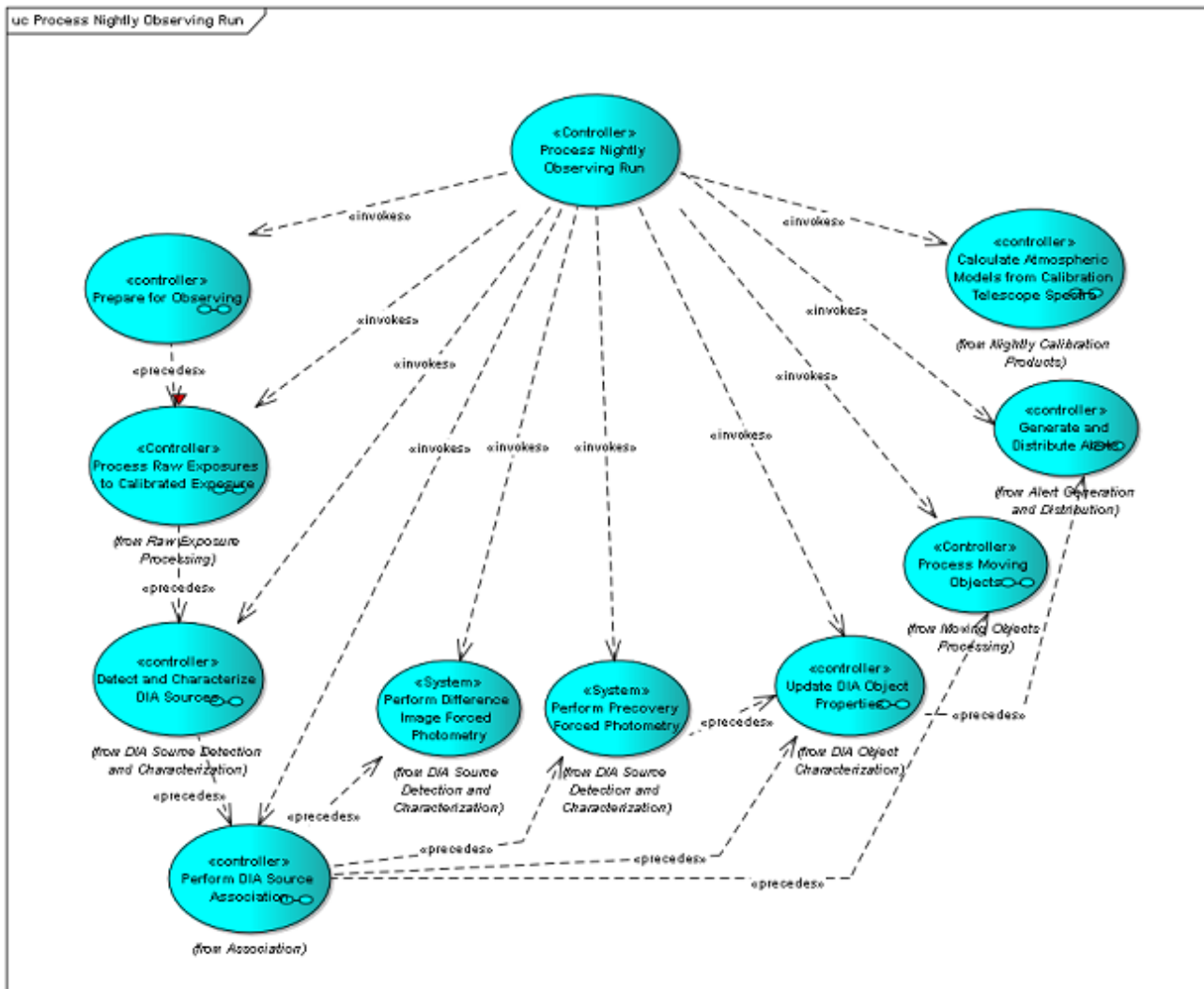
Noćna obrada (engl. *nightly pipeline* ili *alert production pipeline*, Slika 6) jedan je od nekoliko LSST sistema za obradu podataka – za prepoznavanje astronomskih objekata koji su promenili svoj sjaj u odnosu na referentnu sliku. Svaka detekcija astronomskog objekta koja premašuje prag

odnosa signal/šum, $\text{SNR}^{15} > 5$ nakon "oduzimanja" (engl. *difference image analysis*) od referentne slike tretira se kao potencijalno značajan događaj i uključuje se u tok podataka koji se emituje. Svaka notifikacija uključuje paket sa LSST informacijama o izvoru, uključujući fotometriju, koordinate i izrez slike (Graham et al., 2024). Tok će se kontinuirano emitovati sa notifikacijama gotovo u realnom vremenu: u latenciju od 60 sekundi se računa vreme procesiranja i povezivanje sa prethodnim instancama istog astronomskog objekta.

LSST planira da distribuira upozorenja u Apache Avro formatu (pomoću Apache Kafka platforme za emitovanje i kontrolu tokova događaja) sa pridruženim istorijskim podacima starim do 12 meseci, sa približnom veličinom paketa od 82 KB - dodatno, LSST može distribuirati i umanjene (engl. *lite*) pakete, bez istorije i izreza slike. Ukupan protok podataka procenjuje se između 0.2 Gbps i 5 Gbps, što je bio glavni razlog da LSST ograniči broj direktnih korisnika i angažuje spoljne brokere. Nakon što je devet timova podnelo kompletne predloge (2020), sedam je izabrano da prima kompletan tok, a dva preko posrednika. Većina izabranih brokera razvijena je i testirana koristeći tok podataka Zwicky Transient Facility (ZTF (Graham et al., 2019b)).

LSST pravi razliku između diaobject (astronomski objekat) i diasource (jedno merenje objekta), sa odnosom jedan-prema-više. Ova terminologija može varirati u dokumentaciji pojedinačnih brokera (object → alert, locus → alert itd.), ali kardinalnost ostaje ista.

15 SNR – signal-to-noise ratio je odnos koji meri jačinu željenog signala astronomskog objekta sa nepoželjnim smetnjama poput svetlosne prezasićenosti ili greške senzora. Obično se vrednost veća od 5 uzima za prihvatljivu u inicijalnoj detekciji, dok su veće vrednosti poželjne za kvalitetan signal.



Slika 6: Slučaj korišćenja noćnog procesiranja i distribuiranja promenljivih objekata u okviru LSST-a. Izvor: Mario Juric, Robyn Allsman, Jeff Kantor. LDM-134 - LSST Data Management Applications UML Use Case Model

6.2 Pregled LSST brokera

Brokери (termin široko prihvaćen u astronomskoj zajednici (Borne, 2008)), odnosno sistemi za prijem i obradu astronomskih tokova/događaja, namenjeni su radu sa visokofrekventnim tokovima podataka u realnom vremenu.

6.2.1 ALerCE

ALerCE (<https://alerce.science/>, <https://github.com/alercebroker>) - Automatic Learning for the

Rapid Classification of Events - je čileanska platforma za obradu događaja, bogata funkcionalnostima, sa online servisima i Python API-jem. Uključuje obradu u realnom vremenu, agregaciju, cross-match, mašinsko učenje (ML) za klasifikaciju i vizualizaciju (Förster et al., 2021).

ALeRCE primenjuje dve ML metode klasifikacije - Balanced Hierarchical Random Forest (Sánchez-Sáez et al., 2021) nad osobinama svetlosnih krivih i konvolucione neuronske mreže za klasifikaciju izreza slike (Carrasco-Davis et al., 2021). U novijim istraživanjima testirani su i Transformer pristupi (Cabrera-Vives et al., 2024), koji su nadmašili prethodne metode tokom ELAsTiCC kampanje. Za izdvajanje anomalija koristi se model zasnovan na dubokim SVDD/autoenkoderskim mrežama (Perez-Carrasco et al., 2023).

ALeRCE nudi više servisa: veb portal za pretragu objekata sa grafičkim prikazima svetlosnih krivih, klasifikacija i arhivskih podataka; portal za praćenje kandidata za supernove; Python klijent; sistem za upravljanje dodatnim posmatranjima; direktan pristup bazi i Jupyter okruženja.

6.2.2 AMPEL

AMPEL (Nordin et al., 2019) (<https://github.com/ampelproject>, <https://github.com/AmpelAstro>) - *Alice in Modular Provenance-Enabled Land* - je skalabilan softverski okvir u Pythonu za obradu velikih skupova podataka, sa specifičnom primenom na tokove astronomskih notifikacija.

AMPEL je fleksibilan - može obrađivati podatke u realnom vremenu ili podatke iz baza - i proširiv, omogućava korisnicima da dodaju sopstveni kod ili integrišu postojeće alate. Uvodi koncept „kanala“ kao podskupa procesa koji pripadaju korisniku, uz optimizaciju resursa izbegavanjem redundantne obrade. Može se instalirati lokalno ili koristiti preko API-ja uz autorizaciju.

6.2.3 ANTARES

ANTARES (Matheson et al., 2014) (<https://antares.noirlab.edu/>, <https://gitlab.com/nsf-noirlab/csdc/antares/>) - *Arizona-NOIRLab Temporal Analysis and Response to Events System* - razvijan je još od 2014.

Omogućava filtriranje, označavanje, kategorizaciju i klasifikaciju kandidata, kao i definisanje naučnih filtera u Pythonu. Dostupan je putem veb portala i Python klijenta. U ovom sistemu termin „locus“ označava astronomski objekat.

6.2.4 Babamul/BOOM

Babamul (Jegou du Laz et al., 2025) (<https://github.com/babamul/babamul>) je broker koji se razvija na Kalifornijskom institutu za tehnologiju (*Caltech*) i koji se bazira na infrastrukturi *Broker Operations and Optimization Model* (BOOM), koji obezbeđuje hardverski i softverski oslonac sposoban da obradi velike količine podataka u realnom vremenu. Babamul je namenjen korelaciji LSST i ZTF tokova podataka, a planira se organizacija korelisanog toka u kanale dodeljene prema nekoj specifičnoj astronomskoj osobini. Kao poslednji u nizu brokera koji su razvijeni, trenutno nudi Python klijent sa osnovnim operacijama pretraživanja objekata.

6.2.5 Fink

Fink (Möller et al., 2021) (<https://fink-broker.org/>, <https://github.com/astrolabsoftware/fink-broker>) je robustan sistem zasnovan na Apache Spark-u i Kafki. Dizajniran je za obradu podataka pri velikom protoku i analizu astronomskih pojava na različitim vremenskim skalama.

Uvodi "naučne module", dodatne funkcionalnosti (ML klasifikator, kros-mečer, agregacija). Sistem je modularan i prilagodljiv. Dostupan je kroz veb portal, Python API i redistribuirani Kafka tok.

6.2.6 Lasair

Lasair (Smith, 2019) (<https://lasair-ztf.lsst.ac.uk/>, <https://github.com/lsst-uk/lasair-lsst>) je platforma za filtriranje, obogaćivanje i skladištenje upozorenja (Williams et al., 2024).

Omogućuje definiciju detaljnog SQL upita nad kontinuiranim i skladištenim podacima, i koncepte kao što su *watchlist*, *watchmap* i *annotator*. Prva dva koncepta odnose se na spisak odnosno oblast na nebu sa objektima od interesa, čime se pravi filter ulaznog toka, dok se anotatori odnose na eksterne module za analizu podataka. Interni klasifikator Sherlock koristi kontekstualne informacije i algoritam pojačanih stabala odlučivanja (engl *boosted decision tree*; Young, 2023).

6.2.7 Pitt-Google

Pitt-Google (Wood-Vasey, 2024) (<https://github.com/mwvgroup/Pitt-Google-Broker>) koristi infrastrukturne prednosti Google Cloud platforme i omogućava filtriranje, (re)distribuciju i analizu uz veliku fleksibilnost. Podržava različite nivoe obrade putem Pub/Sub servisa i model pretplate,

kao i ugrađenu podršku za HEALpix¹⁶. Poseban fokus je na skalabilnosti i interoperabilnosti sa drugim servisima u okviru Google Cloud sistema.

6.3 Poređenje LSST brokera

6.3.1 Koncepti dizajna

Iako je većina brokera osmišljena i razvijena u poslednjih pet godina, sama ideja i prateći koncepti stari su oko dve decenije. Specijalizovana konferencija „*Hot-wiring the Transient Universe*“, koja se održava od 2007. godine, obuhvata diskusije o brojnim aspektima automatizacije procesa vezanih za podatke o tranzijentnim događajima.

U poređenju sa dizajnom zasnovanim na relevantnoj literaturi iz oblasti obrade događaja (Slika 4), gde plavi pravougaonici predstavljaju interne agente za obradu događaja, a žuti šestougaonici interoperabilne komunikacione kanale, većina brokera i dalje se zasniva na sličnim principima - primeni jednostavnog filtriranja radi sužavanja izbora, obogaćivanju dodavanjem istorijskih podataka (kao što su spektroskopija ili dodatna fotometrija) iz internih ili eksternih baza podataka, primeni metoda mašinskog učenja i/ili unapred definisanih obrazaca, izboru najzanimljivijih kandidata za dodatna posmatranja i konačno redistribuciji izmenjenog toka u istom interoperabilnom formatu podataka. Neki od njih zadržavaju modularnu fleksibilnost koja omogućava preskakanje ili kombinovanje pojedinih koraka.

Donekle u suprotnosti sa prikazom na Slici 4, može se uočiti dominacija metoda mašinskog učenja u odnosu na prepoznavanje paterna / kompleksnih događaja, što se može objasniti opštim trendom razvoja ML/DL metoda u astronomskoj zajednici. Ovaj pomak takođe prati širi trend u nauci o podacima, gde se heuristički modeli sve više zamenjuju statističkim i dubokim modelima treniranim nad velikim skupovima podataka. Ipak, ni zaključci doneseni pomoću ML metoda klasifikacije najčešće nisu konačni i zahtevaju dodatna sukcesivna posmatranja.

Značajna razlika koja utiče na interoperabilnost podataka i servisa je odustajanje LSST-a od VOEvent¹⁷ formata. Ova odluka je doneta pre svega zbog ograničenja u pogledu efikasnosti i

¹⁶ HEALPix (Hierarchical Equal Area isoLatitude Pixelation) predstavlja algoritam za sfernu pikselizaciju koji obezbeđuje jednaku površinu svakog piksela i koristi se za obradu podataka na sferi

¹⁷ VOEvent je standard razvijen u okviru Međunarodne Unije virtualnih opservatorija - *International Virtual*

skalabilnosti u kontekstu izuzetno velikog protoka podataka koji se očekuje u LSST/Rubin eri. VOEvent, zasnovan na XML strukturi, podrazumeva značajan teret u smislu veličine poruke i vremena parsiranja, što postaje kritično kada se razmatraju milioni notifikacija po noći i protoci reda veličine više gigabita u sekundi. U takvom okruženju, potreban je format koji omogućava brzu serijalizaciju i deserijalizaciju, kao i kompaktniji zapis podataka, što je dovelo do izbora Apache Avro formata. Avro koristi binarnu reprezentaciju i šeme podataka, čime se značajno smanjuje veličina poruka i ubrzava njihova obrada u realnom vremenu. Dodatno, binarni formati poput Avro-a bolje se uklapaju u distribuirane sisteme za obradu tokova podataka kao što je Apache Kafka, koji čine osnovu moderne infrastrukture za obradu podataka velikog obima.

Drugi važan razlog odnosi se na fleksibilnost i kompatibilnost sa savremenim sistemima za obradu tokova podataka. LSST je zahtevao format koji podržava evoluciju šeme (schema evolution), verzionisanje i efikasno rukovanje istorijskim podacima unutar istog paketa upozorenja, što Avro prirodno omogućava (ovo je važno za dugotrajne projekte poput LSST-a, gde se struktura podataka može menjati tokom godina bez narušavanja kompatibilnosti sa postojećim sistemima). Istovremeno, prelazak na Avro olakšava implementaciju i održavanje brokerskih sistema, jer omogućava direktnu integraciju sa industrijski standardizovanim alatima za obradu velikih tokova podataka.

Odluku o odustajanju od međunarodnog standarda možemo posmatrati u svetlu realnih potreba velikih naučnih projekata, koji sve više zavise od performansi, skalabilnosti i operativne efikasnosti, naspram standardizacionih tela koja često razvijaju teorijske i opšte okvire sa sporijim ciklusima prilagođavanja. U slučaju LSST-a, zahtevi koje nameće obrada ekstremno velikih tokova podataka u realnom vremenu prevagnuli su nad prednostima interoperabilnosti koje pruža VOEvent standard. Ovo se poklapa sa obrascem u savremenoj nauci i drugim oblastima vezanim za obradu podataka, gde vodeći projekti sve češće usvajaju industrijske tehnologije i prilagođena rešenja kako bi zadovoljili konkretne operativne potrebe, čak i po cenu odstupanja od formalno uspostavljenih standarda.

Takav razvoj sugerše i tenziju između standardizacije i inovacije, gde standardi često zaostaju za praktičnim zahtevima sistema velikog obima, te se naknadno prilagođavaju rešenjima koja su već

Observatory Alliance (IVOA) za razmenu informacija o astronomskim tranzijentnim događajima, zasnovan na XML formatu i namenjen interoperabilnoj distribuciji upozorenja između različitih sistema i institucija.

proverena u praksi.

Dodatna posledica koja se tiče ovog istraživanja je trenutna neusklađenost naziva parametara i metoda brokerskih API-ja.

6.3.2 Funkcionalnosti

Funkcionalnosti različitih brokera mogu se razlikovati ili preklapati, a uobičajeno je i korišćenje različitih termina za iste koncepte. Na primer, AMPEL koristi četvoroslojnu arhitekturu obrade događaja, sa sledećim konceptima – *add*, *combine*, *complement*, *react* – koji odgovaraju obrascima dizajna *filter*, *correlate*, *enrich* i *pattern match*, kako su definisani u literaturi (Tsimelzon, 2006).

Osnovni koncepti obrade događaja (kao i sinonimi koji se koriste u dokumentaciji brokera, navedeni u zagradaama), definisani u ključnom delu iz ove oblasti (Etzion & Niblett, 2010), obuhvataju: prijem, filtriranje, korelaciju (spajanje tokova, *multimessaging*), obogaćivanje (*complement*, *annotate*, *query*, *cross-match*, *xmatch*), uparivanje (mašinsko učenje, *classify*, *characterize*, *react*) i distribuciju. Brokeri često nude i dodatne mogućnosti preuzete iz drugih tipova softvera, kao što su bogati korisnički interfejsi, portfoliji objekata, programabilni i proširivi API-ji itd. Pojedini brokeri time teže da prevaziđu osnovnu ulogu posrednika i prerastu u višenamenske, integrisane astronomske platforme.

Svim brokerima zajednički su osnovni mehanizmi filtriranja tokova, kros-mečer (uz razlike između internog skladištenja i eksternog pristupa) i API-ji koji omogućavaju programsko proširenje ka spoljnim alatima (npr. prilagođeni analitički alati ili TOM/Marshall sistemi). Takođe, svi brokeri su otvorenog koda (engl. *open-source*) i dostupni kroz javne git repozitorijume. Većina brokera je besplatna – Pitt-Google će raditi po modelu pretplate, naplaćujući „cloud“ servise, dok neki (poput AMPEL-a) nude ograničen pristup verziji "uživo" koja se izvršava na njihovoj infrastrukturi.

Većina brokera koristi mašinsko (duboko) učenje za klasifikaciju - pri čemu pravimo razliku između timova koji razvijaju veliki broj detaljnih modela, onih koji nude jedan ili nekoliko osnovnih modela, i onih koji omogućavaju integraciju eksternih modela. Samo Fink i Pitt-Google zasnivaju svoju infrastrukturu na proverenim industrijskim platformama za obradu tokova, dok svi brokeri koriste klasterska rešenja za skladištenje podataka, bilo relacione, dokumentne ili njihove kombinacije. ALERCE, ANTARES, Fink i Lasair razvili su napredne veb portale sa mogućnostima online upita i portfolijima astronomskih objekata, kao i drugim specifičnim funkcionalnostima.

ANTARES, Fink, Lasair i Pitt-Google mogu redistribuirati upozorenja u originalnom Avro formatu.

Iako su zasnovani na različitim principima, može se reći da AMPEL, Fink, Lasair i Pitt-Google dele koncept modularnosti, omogućavajući korisnicima i programerima da izaberu koje će aspekte ili funkcije sistema uključiti u sopstveni radni tok.

6.3.3 Performanse

ELAsTiCC je sedam brokerskih sistema proverio na simuliranom toku od približno 50 miliona upozorenja za oko pet miliona događaja (LSST DESC, bez datuma).

Direktno poređenje performansi nije adekvatno, jer brokeri nisu alternative za isti zadatak, već različita rešenja za različite naučne potrebe.

6.4 Zaključak

Prikazali smo pregled brokera za astronomske tranzijente. Iako dele osnovne funkcionalnosti, razlikuju se u arhitekturi i fokusu.

Opšti zaključci:

- Lasair i ALeRCE imaju najrazvijenije web portale
- ALeRCE ima najrazvijenije ML metode
- Fink i Pitt-Google imaju najrobustnije infrastrukture
- AMPEL nudi najveću fleksibilnost arhitekture

Ukupno, Fink ispunjava sve analizirane zahteve, iako nije dominantan u svakom segmentu. Izbor brokera zavisi od konkretnog naučnog slučaja, dostupnih resursa i daljeg razvoja LSST projekta.

7 Alertissimo: Koncept, namena i opšti pregled sistema

U ovom poglavlju predstavljen je Alertissimo kao sistem za orkestraciju heterogenih alata i tokova podataka u astronomiji vremenskog domena. Najpre se razmatraju motivacija, ciljevi i karakteristični naučni scenariji koji su usmeravali njegovo koncipiranje i projektovanje, a zatim se daje pregled arhitekture i osnovnih slojeva sistema, kao osnova za njihovo detaljnije razmatranje u narednim poglavljima.

7.1 Motivacija za razvoj sistema

Savremena astronomija vremenskog domena zasniva se na posmatračkim sistemima koji proizvode velike količine podataka o promenljivim i prolaznim astronomskim pojavama. Istraživačima su sve češće dostupni kontinuirani ili učestalo ažurirani skupovi detekcija, merenja, klasifikacija i izvedenih podataka. Posebno važnu ulogu u tom okruženju imaju astronomski brokerski sistemi, koji primaju podatke iz velikih pregleda neba, obogaćuju ih dodatnim informacijama i omogućavaju izdvajanje naučno relevantnih kandidata (Graham et al., 2012; Ivezić et al., 2019; Förster et al., 2021; Möller et al., 2021).

U takvom okruženju naučna vrednost ne zavisi samo od tačnosti pojedinačne klasifikacije ili notifikacije, već i od niza odluka o tome koje kandidate treba dodatno proveriti, obogatiti ili proslediti ka narednim posmatranjima. Ovakvo stanovište zastupljeno je i u rukopisu *Toward intelligent, decision-aware systems for LSST-scale astronomy* (Bom et al., 2026), koji se priprema u okviru TVS Science Collaboration¹⁸, gde se LSST posmatra kao dinamično okruženje sa parcijalnim kontekstom, u kojem se odluke donose u uslovima neizvesnosti i ograničenih resursa. Zbog toga predmet integracije više nisu samo pojedinačni tokovi podataka, već i međusobno povezani brokerski, kataloški i analitički servisi koji učestvuju u istom naučnom postupku.

Pojedinačni brokeri, međutim, nisu međusobno zamenljivi. Oni se razlikuju prema izvorima podataka koje podržavaju, dostupnim klasifikatorima, kataloškim ukrštanjima, načinima pretraživanja, vrstama fotometrijskih podataka i mogućnostima pristupa. Neki broker može biti naročito pogodan za početno pronalaženje kandidata, drugi za preuzimanje svetlosnih krivih, treći za pristup rezultatima klasifikacije, a četvrti za podatke o povezivanju sa spoljnim katalozima.

¹⁸ TVS Science Collaboration – naučna saradnja okupljena oko izučavanja astronomskih tranzijenata u kontekstu podataka sa LSST-a i drugih izvora kao i razvoja i analize softvera u zadatom ekosistemu

Funkcionalnosti pojedinačnih brokera uglavnom su dostupne putem njihovih API-ja, pa bi se za unapred definisane scenarije relativno lako mogli napisati namenski skriptovi koji neposredno pozivaju konkretne servise. Problem, međutim, nije sama dostupnost podataka, već odsustvo jedinstvenog pristupa, zajedničkog značenja operacija i podataka, kao i opšteg mehanizma za njihovo povezivanje u složene tokove rada. Cilj Alertissima zato nije izrada još jednog skripta za ograničen broj unapred poznatih slučajeva, već razvoj sistema koji za svaki scenario izraziv u okviru podržanih naučnih operacija može da generiše odgovarajući fizički plan, uskladi različite API-je i reprezentacije i orkestrira njegovo izvršenje nad raspoloživim brokerskim servisima.

Problem nije ograničen na činjenicu da različiti brokeri nude različite funkcionalnosti. Heterogenost se pojavljuje na više nivoa: u nazivima operacija i parametara, strukturama odgovora i unutrašnjoj reprezentaciji podataka, identifikatorima astronomskih objekata, jedinicama mere i transformacijama vrednosti, transportnim protokolima, autentifikaciji, Python klijentima i REST servisima, kao i u značenju pojedinih polja. Čak i kada dva sistema nude prividno istu vrstu podatka, njegov fizički oblik, obim i način dobijanja mogu se značajno razlikovati (Graham et al., 2012; Vujčić et al., 2025; Wiederhold, 1992).

U uobičajenom pristupu teret prevazilaženja tih razlika najvećim delom snosi istraživač. Za svaki servis potrebno je proučiti dokumentaciju, napisati poseban pristupni kod, prilagoditi parametre, obraditi odgovor i prevesti rezultate u oblik pogodan za dalje korišćenje. Kada jedan naučni scenario obuhvata više brokera, potrebno je dodatno preneti identifikatore objekata između koraka, povezati rezultate različitih poziva, voditi računa o poreklu podataka i sačuvati dovoljno informacija da bi postupak mogao da bude ponovljen. Tako nastali programi često su usko vezani za trenutne verzije konkretnih servisa i teško se prenose na druge brokere, izvore podataka ili naučne scenarije (Wiederhold, 1992; Gil et al., 2007; Deelman et al., 2009).

Orkestrator heterogenih brokerskih sistema i tokova podataka Alertissimo (može se popularno i koncizno nazvati i "nad-broker") se razvija kao odgovor na ovaj problem. Njegova osnovna namena jeste da obezbedi jedinstven sloj za izražavanje i izvršavanje složenih astronomskih tokova rada nad heterogenim brokerskim servisima. Sistem ne zamenjuje postojeće brokere i ne pokušava da objedini sve njihove funkcije u još jedan centralizovan servis. Umesto toga, on ih povezuje i koordinira, omogućavajući da se njihove komplementarne mogućnosti koriste u okviru jednog naučnog postupka. Prvobitni koncept Alertissima i njegova uloga u orkestraciji LSST brokerskih

tokova predstavljani su u radu Vujčić et al. (2026).

Implementacija prototipa Alertissimo javno je dostupna u GitHub repozitorijumu projekta (<https://github.com/sambolino/alertissimo>). Verzija 0.1.0, koja odgovara stanju implementacije opisanom i proverenom u ovoj disertaciji, trajno je arhivirana u repozitorijumu Zenodo (doi:[10.5281/zenodo.22688389](https://doi.org/10.5281/zenodo.22688389)). Dalji razvoj Alertissima može se pratiti u GitHub repozitorijumu, dok je verzija opisana u ovoj disertaciji trajno arhivirana na Zenodu i dostupna za citiranje putem DOI oznake.

Najjednostavnije postavljeno, Alertissimo predstavlja posrednički i orkestracioni sloj između istraživača i fizičkih izvora podataka. Naučni zahtev opisuje se nezavisno od pojedinačnih API poziva, dok sistem određuje kako taj zahtev može biti realizovan korišćenjem raspoloživih servisa. Fundamentalni postulat dizajna je razdvajanje pitanja šta istraživač želi da postigne od pitanja kako se odgovarajući podaci fizički dobijaju i obrađuju.

Ovako postavljen orkestracioni sloj ostaje odvojen i od konkretnog korisničkog interfejsa. Isti formalizovani naučni scenario može biti zadat domenski specifičnim jezikom (DSL)¹⁹, u budućnosti sastavljen vizuelno ili preveden iz prirodnog jezika pomoću NLP²⁰/LLM²¹ modula, dok formalna validacija, planiranje i izvršno jezgro ostaju isti. Ovakva separacija odgovornosti oslanja se na principe labave povezanosti (engl. loose coupling) i razdvajanja nadležnosti (engl. separation of concerns) (Dijkstra, 1982; Sommerville, 2011; Martin, 2017).

7.2 Osnovni pristup

Polazna ideja Alertissima jeste da naučni scenario treba da ostane stabilan čak i kada se menjaju unutrašnji tehnički detalji njegovog izvršenja. Istraživač, na primer, može zahtevati pronalaženje objekata u određenom delu neba, izdvajanje kandidata sa određenim osobinama, preuzimanje njihovih svetlosnih krivih i proveru kod više nezavisnih izvora. Takav zahtev ima smisao na nivou astronomskog problema i ne bi trebalo da zavisi od fizičkih naziva parametara koje koriste pojedinačni servisi (Parnas, 1972; Shaw & Garlan, 1996; Ludäscher et al., 2006; Gil et al., 2007).

Alertissimo zato najpre formalizuje korisnički opis u kanonsku reprezentaciju toka rada. Pre

¹⁹ DSL – Domain-Specific Language: jezik prilagođen specifičnoj oblasti primene, često vezanoj za usko specijalizovan domen

²⁰ NLP – Natural Language Processing: oblast koja se bavi obradom i razumevanjem prirodnog / ljudskog jezika

²¹ LLM – Large Language Model: veliki jezički modeli koji se treniraju pomoću mašinskog učenja

izvršenja proverava se da li tražene operacije mogu biti realizovane korišćenjem deklariranih sposobnosti dostupnih brokera. Tek nakon toga formira se fizički plan: biraju se odgovarajući servisi i operacije, kanonski zahtevi prevode se u konkretne parametre, a pozivi se izvršavaju odgovarajućim načinom komunikacije. Ovakav postupak omogućava da jedan naučni korak bude realizovan na različite načine kod različitih brokera. U pojedinim slučajevima jedna fizička operacija može pružiti sve potrebne podatke, dok je kod drugog brokera za isti naučni zahtev potrebno izvršiti više komplementarnih poziva. Odluka o tome ne pripada korisničkom opisu scenarija već planiranju i orkestraciji, što zahteva pažljivo osmišljavanje arhitekture srednjeg sloja.

Odgovori brokerskih servisa zatim se prevode u zajednički oblik pogodan za dalju obradu. Alertissimo pri tome ne svodi rezultate na najjednostavniji zajednički skup polja, jer bi takav pristup doveo do gubitka naučno značajnih informacija. Cilj je da se razlike u fizičkom predstavljanju uklone tamo gde je to opravdano, uz očuvanje specifičnih podataka i njihovog porekla (Wilkinson et al., 2016; Wiederhold, 1992; Bernstein & Melnik, 2007). Podaci povezani sa istim pouzdano identifikovanim objektom organizuju se u centralni semantički proizvod sistema – *Portfolio*. Ovakav koncept u okvirima astronomije vremenskog domena prvi put nalazimo kod Williams et al, 2009 a nešto detaljnije objašnjen u Graham et al, 2012. *Portfolio* se može posmatrati kao strukturisana celina koja se tokom izvršenja može postupno proširivati i koja okuplja podatke o jednom astronomskom objektu. Početni korak može formirati osnovni *Portfolio* na osnovu rezultata pretrage, a naredni koraci mogu ga obogatiti svetlosnim krivama, klasifikacijama, kataloškim ukrštanjima ili drugim raspoloživim informacijama. Na taj način rezultat toka rada nije skup međusobno nepovezanih odgovora različitih API servisa, već akumulirani naučni materijal čiji je identitet utvrđen, podaci korelisani, a poreklo sačuvano.

Automatsko objedinjavanje podataka primenjuje se samo kada postoji pouzdana zajednička identifikacija objekta. Ono se razlikuje od naučnog povezivanja objekata iz različitih pregleda neba, koje zahteva eksplicitnu operaciju i odgovarajući kriterijum, kao što je prostorna blizina.

Pored operacija koje se izvršavaju kod brokerskih servisa, Alertissimo može da primeni i lokalne operacije nad već pribavljenim i uređenim podacima. Deo uslova može biti prosleđen udaljenom servisu ako ga on podržava, dok se preostali deo može proveriti nakon prijema rezultata. Lokalno se mogu izvršavati filtriranje, povezivanje kandidata, potvrđivanje uslova i izvođenje novih proizvoda iz već dostupnog materijala. Rezultat takve operacije može zatim postati ulaz za naredni brokerski

poziv, čime se omogućavaju višestepeni tokovi u kojima se odluke donose na osnovu prethodno pribavljenih podataka.

Važna posledica ovog pristupa jeste da Alertissimo ne služi samo kao prolazni sloj koji prosleđuje zahteve i vraća odgovore. Sistem održava strukturu celog toka rada, vezu između naučnih koraka i fizičkih izvršenja, akumulaciju podataka i informacije potrebne za razumevanje porekla rezultata. Ovo omogućuje reproducibilnost postupka, mogućnost provere rezultata i prenosivost naučnih scenarija između različitih tehničkih okruženja.

7.3 Karakteristični uopšteni slučajevi korišćenja

Radi ilustracije koncepta i namene softverskog sistema Alertissimo, navešćemo ovde nekoliko generalnih primera slučajeva korišćenja. Konkretniji primeri formalnog opisa naučnih scenarija, sa odgovarajućim DSL definicijama, biće prikazani u poglavlju 10.

7.3.1 Pronalaženje kandidata i obogaćivanje kod više brokera

Jedan broker može biti upotrebljen za pronalaženje objekata u zadatom području neba, dok drugi broker pruža pogodniju svetlosnu krivu, a treći dodatne podatke o kataloškim ukrštanjima. U tradicionalnom pristupu istraživač bi morao da napiše poseban programski kod za svaki servis, izdvoji identifikatore iz prvog odgovora i prilagodi ih zahtevima narednih poziva.

Alertissimo ovaj postupak tretira kao jedinstven tok rada. Kandidati pronađeni u prvom koraku postaju ciljevi narednih operacija, dok se novodobijeni podaci pridružuju već postojećem materijalu. Na primer, objekat može biti pronađen korišćenjem servisa ALeRCE ili Lasair, a zatim obogaćen fotometrijskom istorijom iz sistema Fink, koja pored regularnih ZTF merenja može obuhvatiti gornje granice, merenja nižeg kvaliteta i dodatne rezultate Finkovih naučnih modula. Zatim može biti dopunjen spoljnim/istorijskim kataloškim informacijama dostupnim preko sistema ANTARES. Ovakav pristup stvara uslove da se za svaki korak može izabrati servis koji pruža kombinaciju podataka najpogodniju za konkretan naučni scenario. Iako se iza takvog postupka nalazi više fizičkih poziva, za korisnika oni predstavljaju delove jednog naučnog scenarija.

Ovaj uopšteni slučaj korišćenja pokazuje osnovnu vrednost orkestracije: istraživač može da kombinuje najbolje raspoložive mogućnosti više sistema, bez zahteva da jedan broker poseduje sve

potrebne podatke i funkcije.

7.3.2 Potvrđivanje kandidata nezavisnim izvorima

Klasifikacija ili samo prisustvo objekta kod jednog brokera ne mora biti dovoljno za uključivanje kandidata u dalje istraživanje. Naučni postupak može zahtevati da kandidat bude pronađen ili da određeni uslov bude proveren i zadovoljen kod više nezavisnih brokerskih sistema.

Alertissimo može da sprovede takvu proveru kao koordinisan postupak. Kandidati se prosleđuju izabranim brokerima, njihovi odgovori se prevode u uporediv oblik, a zatim se proverava da li je dostignut traženi nivo saglasnosti / kvorum. Tek kandidati koji zadovolje zadati kriterijum nastavljaju ka narednim koracima, na primer ka preuzimanju svetlosne krive ili detaljnijoj analizi. U kompleksnijem scenariju, može se tražiti (parcijalni) kvorum i nad specifičnim semantičkim predikatima, poput klasifikacije. Ovakva vrsta kvoruma zahteva semantičku ujednačenost celokupnog prostora izlaznih atributa ekosistema brokera i drugih aktera.

7.3.3 Povezivanje podataka iz različitih pregleda neba

Istraživanje astronomskih pojava često zahteva zajedničko razmatranje podataka iz različitih pregleda neba, kao što su ZTF i LSST. Objekti u tim sistemima mogu imati različite identifikatore i biti predstavljeni različitim strukturama podataka. Zbog toga tehničko spajanje odgovora nije dovoljno da bi se zaključilo da zapisi predstavljaju isti fizički izvor (Budavári & Szalay, 2008).

Alertissimo omogućava da se kandidati iz više pregleda prvo pribave i uredi, a zatim povežu na osnovu eksplicitno izabranog naučnog kriterijuma – npr. ugaonog rastojanja. Samo kandidati koji zadovolje kriterijum povezivanja mogu učestvovati u toku operacija koji sledi. Time se jasno odvaja tehničko objedinjavanje podataka sa istim identitetom od naučnog zaključivanja o povezanosti objekata iz različitih izvora.

Takav tok rada predstavlja osnovu za buduće složenije analize u kojima se isti događaj ili izvor razmatra kroz više posmatračkih sistema, talasnih dužina ili vrsta astronomskih signala.

7.3.4 Lokalno izdvajanje kandidata i nastavak toka rada

Brokerski servis ne mora podržavati svaki uslov koji je značajan za konkretno istraživanje. Neke

kriterijume moguće je neposredno uključiti u udaljeni upit, dok se drugi mogu proveriti tek nakon što podaci budu pribavljeni i prevedeni u zajednički oblik.

Alertissimo može da kombinuje ova dva načina obrade. Uslovi koje izabrani servis podržava mogu se izvršiti na strani servisa, čime se smanjuje količina prenetih podataka. Preostali uslovi primenjuju se lokalno nad uređenim rezultatima. Kandidati koji zadovolje kriterijume lokalnog filtriranja mogu zatim biti korišćeni kao ciljevi narednog brokerskog poziva.

Na primer, početna pretraga može pronaći širi skup objekata, prvi korak obogaćivanja može pribaviti podatke potrebne za procenu njihovog kvaliteta, a lokalno filtriranje zatim izdvojiti samo odgovarajuće kandidate. Drugi broker tada se poziva samo za taj uži skup – time se fleksibilno dopušta promena skupa kandidata između koraka, umesto da svaki servis nezavisno obrađuje celokupan početni skup.

7.4 Arhitektura sistema

Arhitektura Alertissima, gledano sa najvišeg nivoa apstrakcije, organizovana je oko konceptualnog razdvajanja:

1. Izražavanja naučne namere uz pomoć domenski-specifičnog jezika, uz validaciju leksike i gramatike kao i proveru sposobnosti i dostupnosti servisa i parametara
2. Interne reprezentacije toka orkestracije i fizičkog plana izvršenja
3. Rezultujućih podataka i njihove transformacije, normalizacije i harmonizacije u zajedničku strukturu koja odgovara semantičkom modelu domena

Prikaz arhitekture dat je u vidu dijagrama toka na slici 7. Korisnički skript najpre se proverava prema formalnoj gramatici kao i deklarisanim sposobnostima / softverskim mogućnostima (engl. *capability*) udaljenih servisa i njihovih parametara. Zatim se parsira i prevodi u kanonski tok rada koji zadržava visok nivo apstrakcije i ne sadrži fizičke detalje pojedinačnih servisa. Na osnovu deklarisanih sposobnosti brokera sistem proverava izvodljivost zahteva i formira plan izvršenja.

Tokom realizacije plana, kanonski argumenti povezuju se sa fizičkim parametrima izabranih servisa. Generičke transformacije usklađuju različite reprezentacije, dok izvršni sloj obavlja REST

pozive ili koristi odgovarajuće Python klijente. Odgovori se zatim normalizuju i organizuju u Portfolio strukture. Akumulirani podaci mogu biti lokalno obrađeni ili iskorišćeni za određivanje ciljeva narednih koraka.

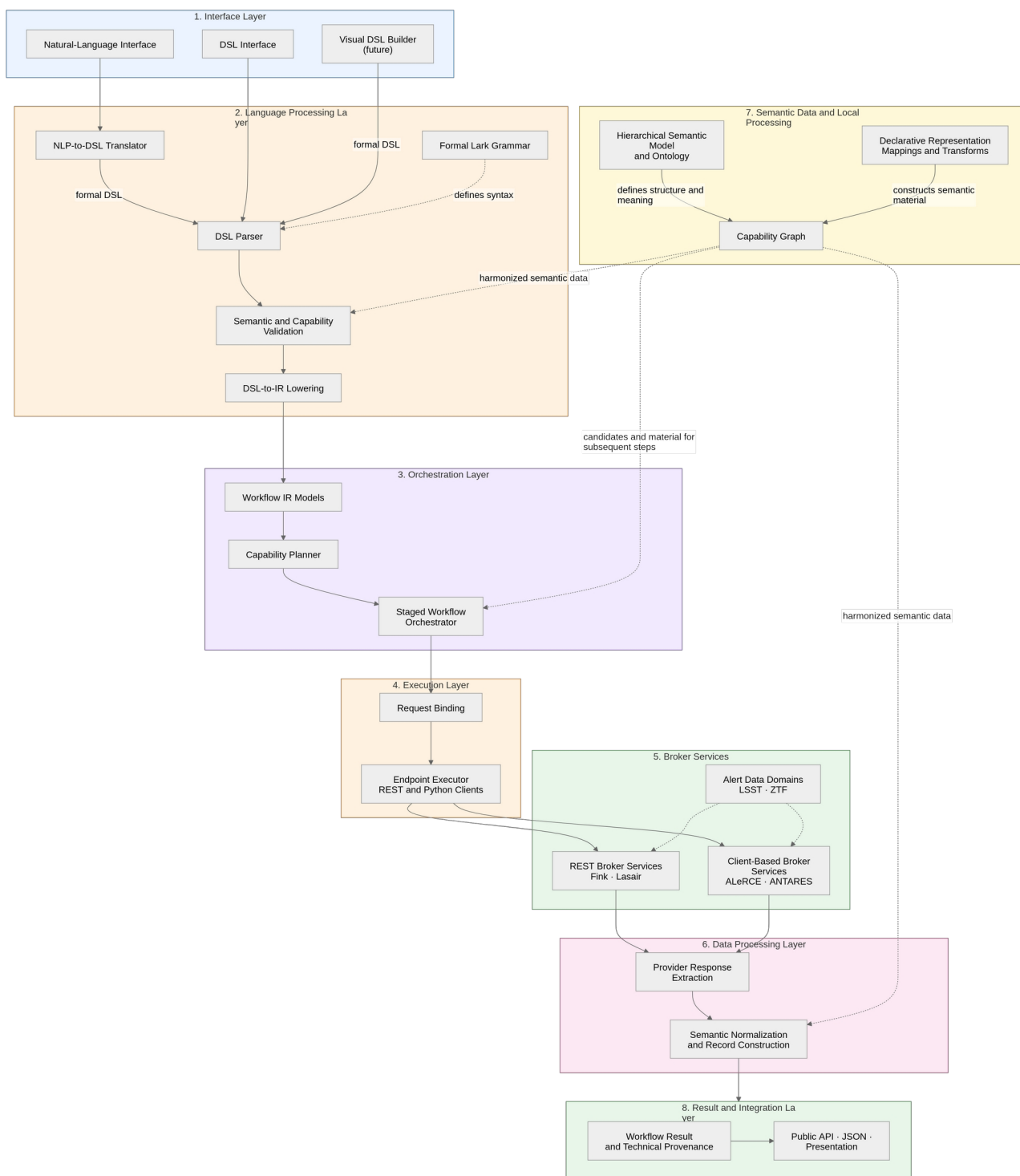
Deklarativni registri imaju važnu ulogu u ovoj arhitekturi. Oni opisuju sposobnosti brokera, fizičke zahteve njihovih servisa i pravila prevođenja ulaznih i izlaznih reprezentacija. Zahvaljujući tome, jezgro orkestracije ne mora da sadrži posebnu proceduralnu logiku za svaki pojedinačni broker (Wiederhold, 1992; Bernstein & Melnik, 2007).

Konačni rezultat obuhvata uređeni naučni materijal, stanje izvršenog toka rada i podatke o provenijenciji. On se može prikazati korisniku, serijalizovati za programski pristup, sačuvati ili proslediti drugom istraživačkom sistemu. Arhitektura tako povezuje korisnički opis problema sa heterogenim servisima, ali pri tome čuva jasne granice između naučne namere, fizičkog izvršenja i interpretabilnog rezultata.

Aktuelna implementacija Alertissima podržava integraciju sistema ALeRCE, ANTARES, Fink i Lasair, za rad sa podacima sa ZTF i LSST teleskopa – što čini ukupno osam obrađenih heterogenih izvora podataka. U sadašnjoj fazi sistem je prvenstveno usmeren na konačne, izvršive tokove zasnovane na zahtevima i odgovorima. Arhitektura predviđa i operacije nad tokovima podataka u realnom vremenu, ali dugotrajni izvršni mehanizmi za kontinuirano praćenje, pretplate, održavanje stanja i reagovanje na nove događaje predstavljaju narednu razvojnu fazu. Ovo proširenje ne zahteva promenu osnovnog principa sistema, već nadogradnju postojećeg orkestracionog jezgra izvršnim okruženjem za kontinuirane tokove. Bitno je naglasiti i da, za razliku od srodnih softverskih rešenja (poput SkyPortal; van der Walt et al, 2019.), Alertissimo semantički planira heterogene realizacije umesto da odvojeno i pojedinačno skriptuje API pozive.

Polazna hipoteza za koncipiranje sistema nad-brokera kao što je Alertissimo pretpostavila je da je moguće uopštiti gradivne blokove takvog sistema tako da oni mogu univerzalno da reprezentuju korake naučnih tokova/scenarija. Takav pristup je dao dobru početnu osnovu, razgraničio slojeve i usmerio projektovanje sistema po principima visoke kohezije i labave sprege. Međutim, pri susretu sa prvim pokušajima implementacije jednostavnijeg prototipa pokazalo se da je domenski atributski prostor ogroman, da postojeći brokeri ne teže zajedničkoj, univerzalnoj reprezentaciji podataka te da se njihova struktura i semantika prvo moraju u potpunosti savladati i razumeti. U praksi je to značilo da se mora proučiti i modelovati celokupan atributski prostor jer parcijalno, prototipsko

modelovanje ne oslikava dovoljno kompleksnost i heterogenost semantičkih sadržaja. Zatim se pokazalo da je ta semantika hijerarhijski slojevita iako je često spljoštena u brokerskim odgovorima. Oživljavanje semantičke višedimenzionalnosti i osmišljavanje hijerarhijskih struktura koje se mogu replicirati bio je jedan od najvećih izazova ovog rada, ali je, kada je na taj problem odgovoreno, on dao usmerenje za modelovanje ostatka sistema. Zato u daljem prikazu rada prvo predstavljamo osnovni semantički prostor domena i njegovo modelovanje.



Slika 7: Dijagram toka Alertissimo sistema za orkestraciju astronomskih tranzijenata. Napomena: ovo je dijagram visokog nivoa apstrakcije, sa okvirnim opisima i vezama. Detalji implementacije pojedinačnih modula biće prikazani u narednim poglavljima.

8 Ontologija i hijerarhijski semantički model

Prethodno poglavlje predstavilo je Alertissimo kao sistem koji konceptualno razdvaja opis naučnog scenarija/toka od fizičke realizacije nad heterogenim brokerskim servisima. Takvo razdvajanje nije moguće samo uvođenjem zajedničkog programskog interfejsa. Potrebna je stabilna osnova koja određuje šta svaki pojedinačni podatak znači, gde pripada u zajedničkoj strukturi, kako se razlikuju višestruke tvrdnje o istom objektu i na koji način se čuva njihovo poreklo. Tu ulogu u Alertissimu imaju ontologija i hijerarhijski semantički model.

Semantički sloj istovremeno predstavlja kanonsku strukturu podataka (u koju se prevode odgovori različitih servisa) i sažetu formalizaciju pojmova, odnosa, kardinalnosti i konteksta koji su potrebni za njihovo jednoznačno tumačenje (Gruber, 1993). Zbog te dvostruke uloge isti model povezuje normalizaciju, validaciju, planiranje, lokalnu obradu, akumulaciju materijala u Portfolio strukturi i semantičku provenijenciju. U aktuelnoj implementaciji njegova specifikacija se nalazi u datoteci ontology.yaml, dok parseri, indeksi i validatori dopunjuju njenu operativnu funkciju.

Ovo poglavlje zato ne razmatra način na koji korisnik zadaje tok rada, već unutrašnji jezik značenja na koji se različiti korisnički i programski interfejsi mogu osloniti. Razlika je važna: interfejs opisuje šta korisnik želi da postigne, dok ontologija određuje koje strukture i pojmovi postoje u sistemu i kako se fizički podaci prevode u njih. Ta stabilna mašinski čitljiva osnova važna je i za buduće prirodno-jezičke i vizuelne interfejse, jer fleksibilan ulaz može biti prihvaćen tek nakon prevođenja u formalno proverljivu reprezentaciju (Vujčić et al., 2026).

Potreba za ovakvim modelom postaje posebno vidljiva u multi-mesindžer astronomiji. Jedan fizički događaj može ostaviti trag u elektromagnetnom zračenju, gravitacionim talasima, neutrinima ili drugim kanalima, ali ti tragovi ne dolaze kao različite kolone iste tabele. Oni nastaju u različitim instrumentima i vremenskim režimima, imaju različite oblike lokalizacije, jedinice, modele greške, pragove detekcije i stepene obrade. Zajednički naučni objekat zato ne ukida razlike između heterogenih podataka: on predstavlja okvir u kojem se njihove međusobno nejednake tvrdnje mogu povezati i zajedno tumačiti (Mészáros et al., 2019).

Elektromagnetno opažanje najčešće se lokalizuje tačkom ili malom oblašću na nebu i može sadržati fotometriju u više filtera, sliku, spektar ili vremensku seriju. Događaj gravitacionih talasa tipično

dolazi sa mapom verovatnoće položaja izvora signala na široj oblasti neba, konturama pouzdanosti, procenom udaljenosti i parametrima izvora izvedenim iz detektovanog signala. Neutrinski događaj može sadržati rekonstruisani smer, ugaonu neizvesnost, energiju i verovatnoću da je događaj astrofizičkog porekla. Naknadna radio, optička, ultraljubičasta, rendgenska ili gama opažanja zatim proizvode nove serije merenja. Čak i kada se sva opažanja odnose na istog kandidata, njihova unutrašnja struktura i epistemološki status nisu isti.

Ravan model bi ovu raznovrsnost morao da predstavi velikim skupom slabo povezanih polja, od kojih bi većina bila prazna za svaki pojedinačni zapis. Isti naziv poput flux, significance ili error mogao bi da označava neuporedive veličine. Hijerarhija umesto toga uvodi kontekst pre vrednosti: najpre se određuje vrsta semantičkog zapisa ili proizvoda, zatim proizvođač i isporučilac (engl *producer, channel*), potom metoda merenja, a tek na najnižem nivou vrednost, greška, jedinica, indikatori kvaliteta itd. U tom kontekstu, značenje ne zavisi od naziva završnog polja već od čitave putanje kroz model.

Veza između heterogenih izvora podataka zato se ne uspostavlja prostim spajanjem njihovih merenja ili izravnavanjem strukture, kao što je čest slučaj u strukturama podataka koje vraćaju brokerski API-ji. Ona se predstavlja eksplicitnom asocijacijom između samostalnih zapisa, uz kriterijume kao što su vremenska blizina, presek oblasti lokalizacije, kompatibilnost udaljenosti ili rezultat naučnog modela. Jedan događaj može imati više optičkih kandidata unutar mape gravitacionih talasa, a svaki kandidat više detekcija, klasifikacija i kataloških veza. Hijerarhija čuva ove kardinalnosti: događaj, kandidat, opažanje, proizvod i element proizvoda nisu isti nivo i ne treba ih svoditi na isti tip zapisa.

Takva organizacija ima i neposrednu softversku posledicu. Nova vrsta merenja može se dodati kao nova grana ili podgrana, bez proširivanja svake postojeće strukture nepovezanim poljima. Zajednički delovi mogu se referencirati i ponovo koristiti, dok specifični podnivoi ostaju dostupni samo tamo gde imaju značenje. Validacija tada proverava pravila odgovarajuće grane, a orkestracija može da radi nad zajedničkim konceptima - vremenom, lokalizacijom, kandidatom ili proizvodom - bez pretpostavke da svi izvori imaju istu fizičku reprezentaciju.

Implementacija razmatrana u ovoj disertaciji ograničena je na ZTF i LSST optičke podatke. Multi-mesindžer primeri zato ne predstavljaju tvrdnju da su svi navedeni izvori već povezani sa izvršnim slojem Alertissima. Njihova uloga je da pokažu zahtev koji je uticao na oblik modela: proširenje na

gravitacione talase, neutrine, radio ili visokoenergetska opažanja treba da dodaje nove semantičke grane i mapiranja, a ne da menja osnovnu organizaciju identiteta, zapisa, kolekcija, veza i provenijencije.

8.1 Uloga semantičkog sloja

Heterogenost brokerskih sistema ne potiče samo iz različitih naziva krajnjih tačaka i parametara. Isti fizički naziv može imati različito značenje, ista naučna veličina može biti zapisana u različitim jedinicama ili strukturama, a pojedini rezultat može predstavljati neposredno merenje, izvedenu vrednost, klasifikacionu tvrdnju ili vezu sa spoljnim katalogom. Unifikacija na nivou transporta zato bi rešila samo način pozivanja servisa, ali ne i interpretaciju njihovih odgovora (Wiederhold, 1992; Graham et al., 2012; Gruber, 1993).

Alertissimo uvodi semantički sloj kao posrednika između fizičkih reprezentacija i orkestracije. Ontologija opisuje dozvoljene pojmove i strukture; deklarativna mapiranja navode gde se odgovarajuće vrednosti nalaze u konkretnom odgovoru; generičke transformacije usklađuju reprezentacije; normalizacija konstruiše semantičke zapise; *Portfolio* ih objedinjuje oko identiteta objekta; a graf softverskih mogućnosti (engl. *capability graph*) – implementiran kao statički registar - iz istih deklaracija izvodi znanje potrebno za validaciju i planiranje. Time se izbegava da jezgro orkestracije poznaje posebne detalje svakog brokera.

Praktični rezultat je uniforman izlaz, koji i sam postaje deo kompleksnije celine toka rada. Naučni materijal pribavljen u jednom koraku može biti bezbedno zadržan, dopunjen podacima drugog izvora, lokalno analiziran i upotrebljen za formiranje narednog koraka. Pri tome ostaje moguće razlikovati ko je proizveo određenu tvrdnju, kojim servisom je ona dostavljena i u kom konkretnom izvršenju je postala deo rezultata.

Tabela 7: Matrica heterogenosti fizičkih realizacija svetlosne krive. Ista semantička operacija može fizički biti eksplicitan lightcurve poziv, objektna istorija, niz alertova ili kompozicija više poziva; orkestracija mora da prevaziđe upravo tu razliku. Zeleno označava direktnu fizičku podršku, a žuto posrednu realizaciju u Alertissimu.

Matrica heterogenosti fizičkih poziva i semantičkih rezultata — PRIMER: lightcurve / svetlosna kriva				
Broker / pregled	Fizička podrška i konkretan poziv	Fizička struktura odgovora	Sadržaj odgovora	Semantičko značenje i normalizacija u Alertissimu
ALeRCE LSST	Status: Direktna - eksplicitna lightcurve metoda Poziv: Python klijent: Alerce.query_lightcurve(oid=<diaObjectId>, survey="lsst"). Optimizator može dodati opcioni Alerce.query_forced_photometry(..)	Oblik: Rečnik (bundle) sa nizovima detections[], forced_photometry[] i non_detections[]. Opcioni forced poziv vraća zaseban niz rečnika.	Detekcije: DA Nedetekcije: deklarirana grana; audit primer je prazan Forced: DA, i u bundle-u i kroz posebnu metodu	Tumačenje: detections su pojedinačna diaSource merenja; forced_photometry su diaForcedSource merenja. Normalizacijom nastaju tačke svetlosne krive, forced_photometry_points i odgovarajući zapisi sa provenijencijom. Primer: Audit primer: 16 detekcija, 10 forced merenja, 0 nedetekcija.
ALeRCE ZTF	Status: Direktna - eksplicitna lightcurve metoda Poziv: Python klijent: Alerce.query_lightcurve(oid=<ZTF objectId>, survey="ztf"). Za potpuniju krivu optimizator dodaje opcioni query_forced_photometry(...).	Oblik: Rečnik sa detections[] i non_detections[]. Forced fotometrija nije u ovom bundle-u, već stiže kao zaseban niz iz query_forced_photometry.	Detekcije: DA Nedetekcije: DA, zaseban niz Forced: DA, ali poseban poziv	Tumačenje: Detekcije nose izmerene magnitude; nedetekcije nose vreme, filter i graničnu magnitudu; forced redovi su nezavisna merenja. Alertissimo ih spaja u jednu svetlosnu krivu sa odvojenim forsiranim tačkama. Primer: Audit primer: 1.044 detekcije, 319 nedetekcija i 286 forced merenja iz dopunskog poziva.
ANTARES LSST	Status: Posredna realizacija u Alertissimu - nema eksplicitnog lightcurve poziva Poziv: Python klijent: antares_client.search.get_by_lsst_dia_object_id(lsst_object_id). Fizički object-history poziv registrovan je i za lightcurve_lookup; zaseban lightcurve poziv nije potreban.	Oblik: Objekat Locus; istorijski redovi su u alerts[] (Alert objekti), sa vrednostima u alert.properties[...]. Locus.lightcurve ostaje sekundarna, duplirana reprezentacija i ne koristi se kao ulaz normalizacije.	Detekcije: DA, u alerts[] Nedetekcije: NE kao posebna grana Forced: Pridružena science/template flux polja postoje u diaSource alertu; nema nezavisnog niza forced istorije	Tumačenje: Alertissimo koristi istoriju iz alerts[] i konstruiše jedan lightcurve@lsst:antares zapis. Svaki diaSource alert postaje jedna tačka u points; pridružena science/template flux polja ne proglašavaju se za nezavisne forced_photometry_points. Time se dobija svetlosna kriva bez oslanjanja na duplirani Locus.lightcurve atribut. Primer: 16 LSST alert redova -> 16 points; jedan združeni lightcurve zapis i bez forced_photometry_points.
ANTARES ZTF	Status: Posredna realizacija u Alertissimu - nema eksplicitnog lightcurve poziva Poziv: Python klijent: antares_client.search.get_by_ztf_object_id(ztf_object_id). Isti object-history endpoint oglašava lightcurve_lookup.	Oblik: Objekat Locus sa alerts[]; svaka Alert instanca ima alert_id, mjd i properties[...]. Kandidati i gornje granice nalaze se u istom nizu i razlikuju se preko properties.ant_survey.	Detekcije: DA, ant_survey=1 Nedetekcije / gornje granice: DA, ant_survey=2 Forced: NE	Tumačenje: Alertissimo iz istog alerts[] niza konstruiše jedan lightcurve@ztf:antares zapis. Redovi sa ant_survey=1 postaju merene tačke, a redovi sa ant_survey=2 tačke sa upper_limit=true i sačuvanom graničnom magnitudom. Isti fizički odgovor i dalje daje i pojedinačne semantičke detekcione zapise. Primer: 316 tačaka - 70 detekcija i 246 gornjih granica

Tabela 7 (nastavak)

Matrica heterogenosti fizičkih poziva i semantičkih rezultata — PRIMER: lightcurve / svetlosna kriva				
Broker / pregled	Fizička podrška i konkretan poziv	Fizička struktura odgovora	Sadržaj odgovora	Semantičko značenje i normalizacija u Alertissimu
Fink LSST	Status: Posredna realizacija u Alertissimu - nema eksplicitnog lightcurve poziva Poziv: REST GET /api/v1/sources? diaObjectId=...; opciono GET /api/v1/fp? diaObjectId=... za potpunu forced istoriju.	Oblik: Oba odgovora su JSON nizovi ravnih rečnika. sources[] sadrži pretežno r: (Rubin) i f: (Fink) ključeve; fp[] sadrži zasebne diaForcedSource redove.	Detekcije: DA, sources[] Nedetekcije: NE kao posebna istorija Forced: DA, fp[]; neka science/template flux polja postoje i u source redovima	Tumačenje: sources nije samo lista objekata već niz diaSource merenja i zato je primarni izvor za rekonstrukciju svetlosne krive. fp daje atomsku forced fotometriju, uključujući epohe koje se ne moraju poklapati sa diaSource epohama. Normalizacija gradi points i forced_photometry_points. Primer: 16 source redova i 20 forced redova; 12 epoha je zajedničko, 4 postoje samo u sources, a 8 samo u fp.
Fink ZTF	Status: Posredna realizacija u Alertissimu - nema eksplicitnog lightcurve poziva Poziv: REST GET /api/v1/objects? objectId=...; za gornje granice koristi se withupperlim=true.	Oblik: JSON niz ravnih rečnika — uprkos imenu objects, elementi su alert/istorijski redovi, ne omotači objekata. Polja imaju prefikse i:, d: i v:; d:tag označava valid, upperlim ili badquality.	Detekcije: DA, d:tag=valid Nedetekcije / gornje granice: DA samo uz withupperlim=true Forced: NE	Tumačenje: valid redovi nose fotometrijsko merenje; upperlim redovi nose limit i ne smeju se tumačiti kao izmerena magnituda; badquality se ne prihvata ni kao detekcija ni kao gornja granica. Rezultat se ujednačava u tačke svetlosne krive. Primer: sa withupperlim=true: 33 reda — 14 valid i 19 upperlim.
Lasair LSST	Status: Posredna realizacija u Alertissimu - nema eksplicitnog lightcurve poziva Poziv: REST GET /api/object/ sa objectId=<diaObjectId>. Fizički object endpoint vezuje target_id i oglašava i object_lookup i lightcurve_lookup.	Oblik: Ugnežđen JSON rečnik: diaObject{...}, diaSourcesList[], diaForcedSourcesList[] i lasairData{...}. Detekcije i nezavisna forced merenja eksplicitno su razdvojeni u dva niza.	Detekcije: DA, diaSourcesList[] Nedetekcije: NE kao posebna grana Forced: DA, diaForcedSourcesList[]	Tumačenje: Naziv fizičkog poziva ostaje object, ali capability i mapiranja sada dokazuju da njegov odgovor realizuje GetLightcurve. Alertissimo gradi jedan lightcurve@lsst:lasair zapis: diaSourcesList[] postaje points, a diaForcedSourcesList[] odvojeni forced_photometry_points. Primer: 245 diaSource redova -> 245 points i 345 diaForcedSource redova -> 345 forced_photometry_points.
Lasair ZTF	Status: Direktna - eksplicitan lightcurve endpoint Poziv: REST POST /api/lightcurves/ sa objectIds=<jedan ili CSV lista ZTF objectId vrednosti>.	Oblik: JSON sloj omotača: [{objectId: ..., candidates: [...]}]. U candidates[] su pomešane detekcije i redovi koji predstavljaju samo graničnu magnitudu.	Detekcije: DA, red ima candid i izmerenu fotometriju Nedetekcije / limiti: DA, red bez candid sa diffmaglim Forced: NE	Tumačenje: Detekcioni redovi postaju merene tačke; redovi sa graničnim vrednostima zadržavaju vreme/filter i photometry.<filter>.limit.mag, bez unošenja PSF magnitude koja nije izmerena. Trenutno se ne postavlja poseban upper_limit indikator kada ga fizički odgovor eksplicitno ne daje. Primer: 92 reda — 35 detekcija i 57 limit-only redova.
Napomena: Direktna fizička podrška označava eksplicitan lightcurve poziv ili metodu brokera. Posredna realizacija označava da Alertissimo ostvaruje GetLightcurve mapiranjem objektne, source ili alert istorije; semantička operacija je zato podržana i kada je fizički API ne izlaže pod tim imenom. „Nedetekcija“ i „gornja granica“ nisu svuda isti fizički tip reda; tabela zato navodi i način kodiranja.				

8.2 Dvostruka uloga modela: struktura i ontologija

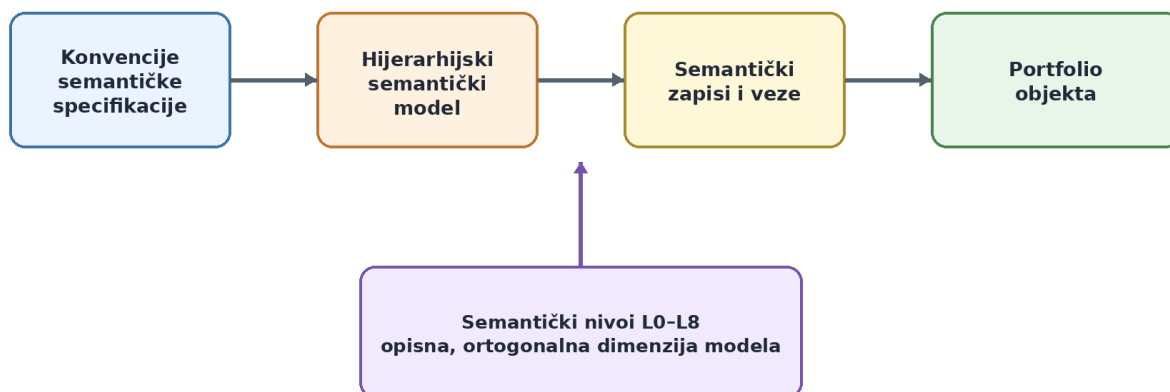
Sa softversko-inženjerskog stanovišta, hijerarhijski semantički model ima dve međusobno povezane uloge. Prva je neposredna i praktična: model predstavlja ponovo upotrebljivu kanonsku strukturu podataka nezavisnu od formata udaljenog servisa. Druga je ontološka: njime se eksplicitno određuju vrste semantičkih zapisa, njihovi delovi, dozvoljene veze i pravila kompozicije. Model zato nije samo DTO, normalizovani JSON ili skup programskih klasa, iako se njegove instance mogu serijalizovati u takve oblike.

Korisna interpretacija može se dati kroz M0–M3 nivoe modelovanja. Ova podela ne predstavlja tvrdnju da je Alertissimo realizovao potpuni MOF ekosistem, već analitički okvir za precizno razdvajanje pravila notacije, konkretnog modela i podataka. Deo metamodelskog značenja trenutno je sadržan u specifikaciji a deo ostaje u implementaciji parsera i validatora. Zbog toga je preciznije govoriti o metamodelskom jezgru nego o potpuno samodefinišućem metamodelu (Object Management Group, 2016).

Nivo	Uloga	Alertissimo realizacija
<i>M3</i>	<i>Opšti mehanizmi za definisanje jezika modelovanja</i>	<i>Pravila parsera, apstraktne sintakse i validacije</i>
<i>M2</i>	<i>Konstrukcije kojima se opisuje model</i>	<i>Kontejneri, polja, reference, kardinalnost, nivoi i veze</i>
<i>M1</i>	<i>Konceptualni model astronomskih podataka</i>	<i>Vrste zapisa, hijerarhijske strukture i dozvoljeni odnosi</i>
<i>M0</i>	<i>Konkretno instance i vrednosti</i>	<i>Semantički zapisi, veze i Portfolio sadržaj u izvršenom toku</i>

Tabela 8: M0–M3 interpretacija semantičkog modela Alertissima. Na M2 nivou nalaze se konvencije za opis kontejnera i polja, jednostruke i višestruke reference, interne ponovo upotrebljive projekcije, parametarski segmenti, semantički nivoi i posebna ravan veza između kontejnera. Na M1 nivou tim konstrukcijama se opisuje konkretan skup astronomskih pojmova, dok M0 čine stvarni zapisi i njihove vrednosti. M3 je u aktuelnom sistemu pre svega sadržan u opštim pravilima parsera i strukturi implementacije. Ovako postavljena granica omogućava punu formalizaciju.

Od pravila modelovanja do podataka



Meta-nivoi M0-M3 i semantički nivoi L0-L8 predstavljaju dve različite ose.

Slika 8: Odnos konvencija semantičke specifikacije, hijerarhijskog modela, konkretnih zapisa i Portfolio reprezentacije. Semantički nivoi L0-L8 predstavljaju posebnu opisnu i diskriminacionu osu i ne treba ih mešati sa M0-M3 nivoima modelovanja.

8.3 Organizacija semantičkog prostora

Specifikacija `ontology.yaml` polazi od osnovnih semantičkih elemenata, a ne od strukture ili rasporeda polja u odgovoru nekog konkretnog servisa. Kontejneri kao što su `position`, `time`, `identity` i `photometry` definišu se jednom, a zatim se od njih sastavljaju zapisi kao što su `summary`, `detection`, `classification` i `lightcurve`. U nastavku se elementi modela uvode tim redom: od kontejnera i putanja do složenih zapisa, kolekcija i veza.

Specifikacija je zapisana u uređenoj notaciji nalik YAML-u. Redosled deklaracija i ponovljene direktive imaju značenje, pa se ona ne može tumačiti kao običan YAML rečnik u kojem se ponovljeni ključevi prepisuju.

8.3.1 Kontejneri, polja i putanje

Ime semantičkog kontejnera piše se unutar izlomljenih zagrada, dok se ime polja piše unutar četvrtastih zagrada. Kontejner uvodi kontekst, a polje označava vrednost koja se može pojaviti u

tom kontekstu. Definicija položaja u aktuelnoj ontologiji, skraćena na dva osnovna polja, glasi:

```
<position>:  
  level: L0, L1  
  unit: deg  
  [ra]:  
  [dec]:
```

Iz nje se izvode semantičke putanje koje jednoznačno imenuju rektascenziju i deklinaciju u okviru položaja:

```
position.ra  
position.dec
```

Ove putanje nisu nazivi kolona udaljenog servisa. Fizičko polje može se zvati `ra`, `meanRa`, `coord_ra` ili drugačije, dok ga deklarativno mapiranje povezuje sa stabilnom semantičkom putanjom. Takva organizacija odgovara ulozi kontrolisanog astronomskeg rečnika, ali čuva i strukturni kontekst pojma (Derriere et al., 2005; Lemson et al., 2018).

8.3.2 Parametarski kontejneri i dinamička polja

Kada jedan segment putanje dobija vrednost tek za konkretan podatak, a ispod sebe i dalje ima poznatu strukturu, koristi se parametarski kontejner. U fotometriji je takav segment oznaka filtra:

```
<photometry>:  
  <{filter}>:  
    <psf>:  
      [mag]:  
      [flux]:  
photometry.g.psf.mag  
photometry.r.psf.mag  
photometry.i.psf.flux
```

Oznaka `{filter}` nije doslovan deo putanje. Pri konkretizaciji je zamenjuje vrednost kao što su `g`, `r` ili `i`, a ta vrednost postaje deo značenja jer merenja u različitim filtrima nisu međusobno zamenljiva.

Za attribute specifične za izvor, koji još nemaju stabilno mesto u modelu, predviđeno je dinamičko polje `[{field}]`:

```

<quality>:
  [real_bogus]:
    type: float
  [{field}]:
    description: 'Additional quality attribute'
quality.real_bogus
quality.broker_score
quality.pipeline_flag

```

Dinamičko polje je kontrolisano proširenje na kraju poznatog konteksta. Ono ne pretvara čitavu strukturu u proizvoljan rečnik, a često korišćen atribut kasnije može dobiti sopstvenu kanonsku definiciju.

8.3.3 Ugnežđivanje

Kontejneri i polja mogu biti ugnežđeni. Svaki novi segment sužava značenje vrednosti, pa nema potrebe da se kontekst ponavlja u samom nazivu polja. Deo fotometrijske definicije za aperturski fluks izgleda ovako:

```

<photometry>:
  <{filter}>:
    <aperture>:
      [flux]:
        unit: nJy
      [error]:
        unit: nJy
photometry.g.aperture.flux
photometry.g.aperture.flux.error

```

Polje `flux` ovde dobija značenje iz cele putanje: reč je o fotometriji u filtru `g`, izmerenoj unutar aperture. Putanja `photometry.g.psf.flux` označava drugu metodu merenja i zato ostaje različita i kada su tip i jedinica vrednosti isti.

8.3.4 Tipovi i ograničenja

Polje može imati opis, tip, jedinicu, dozvoljeni opseg ili skup kontrolisanih vrednosti. Specifikacija ne određuje samo ime podatka već i osnovne uslove njegovog tumačenja:

```

<quality>:

```

```
[real_bogus]:  
  type: float  
  range: [0, 1]
```

<detector_status>:

```
[status]:  
  enum: [on, off, triggered, unknown]
```

Broj 1.4 jeste sintaksno ispravna realna vrednost, ali nije dozvoljena vrednost za `quality.real_bogus`. Slično tome, status detektora mora pripadati navedenom skupu. Ova pravila pripadaju semantičkoj specifikaciji; obim njihove automatske provere u aktuelnom prototipu razmatra se na kraju poglavlja.

8.3.5 Semantički zapisi i kvalifikatori porekla

Dosadašnji primeri opisivali su delove modela koji se mogu ponovo koristiti. Normalizovani rezultat, međutim, nosi vrstu zapisa koja određuje kakvu naučnu celinu predstavlja. U aktuelnom modelu to mogu biti, između ostalog, sažetak objekta (`summary`), pojedinačna detekcija ili događaj (`detection`), klasifikacija (`classification`), kataloško povezivanje (`crossmatch`), svetlosna kriva (`lightcurve`), spektar (`spectrum`) ili drugi podatkovni proizvod (`data_product`).

Definicija `detection` uvodi zapis jedne opservacije, detekcije, alerta ili događaja. Njeno početno polje razlikuje osnovne vrste fizičkog sadržaja:

<detection>:

```
description: "Single detection, observation, alert, or event record"  
[type]:  
  enum: [optical, infrared, ultraviolet, xray, gamma_ray,  
         radio, solar_system, gravitational_wave, neutrino,  
         satellite, multimessenger, other]
```

Ostatak zapisa `detection` sastavlja se od položaja, vremena, identiteta, kvaliteta, provenijencije i sadržaja svojstvenog pojedinim glasnicima. Ti delovi se ne prepisuju u definiciju zapisa, već se uključuju kompozicijom koja je objašnjena u odeljku 8.3.7. U izvršnom modelu konkretna instanca predstavljena je objektom klase `SemanticRecord`; ona čuva vrstu zapisa i polja navedena relativno prema toj vrsti.

Vrsta zapisa može biti dopunjena proizvođačem i kanalom kojim je zapis dostavljen. Opšti oblik

kvalifikatora je `record@producer:channel`, pri čemu je kanal opcioni deo:

`detection@ztf:fink`

`detection@lsst:antares`

`classification@fink`

`crossmatch@gaia:fink`

U oznaci `detection@ztf:fink` pregled ZTF je proizvođač detekcije, a Fink kanal preko kojeg je ona dobijena. Oznaka `classification@fink` predstavlja klasifikacionu tvrdnju koju proizvodi Fink, dok `crossmatch@gaia:fink` označava rezultat zasnovan na katalogu Gaia koji je dostavljen preko Finka. Kvalifikator pripada zapisu i ne ponavlja se ispred svakog ugnežđenog polja; potpuna putanja zato može glasiti `detection@ztf:fink.photometry.g.psf.mag`.

8.3.6 Semantički nivoi

Nivoi L0–L8 opisuju kontekst u kojem se pojam koristi. Oni ne predstavljaju dubinu ugnežđivanja, nisu deo javne putanje i ne odgovaraju nivoima modelovanja M0–M3. Njihova uloga je da razlikuju, na primer, agregat objekta od pojedinačnog merenja ili vremensku seriju od klasifikacione tvrdnje.

Nivo	Tipičan kontekst	Primer uloge
<i>L0</i>	<i>Sažetak i identitet objekta</i>	<i>Identifikatori i agregatne osobine</i>
<i>L1</i>	<i>Pojedinačno merenje</i>	<i>Detekcija i neposredni metapodaci</i>
<i>L2</i>	<i>Specifični fenomenološki sadržaj</i>	<i>Posebne vrste fizičkih događaja</i>
<i>L3</i>	<i>Specifični proizvod podataka</i>	<i>Slika, spektar ili binarni sadržaj</i>
<i>L4</i>	<i>Vremenske serije</i>	<i>Svetlosne krive i izvedene osobine</i>
<i>L5</i>	<i>Procena i klasifikacija</i>	<i>Kvalitet i izlazi modela mašinskog učenja</i>
<i>L6</i>	<i>Obogaćivanje i povezivanje</i>	<i>Spoljni katalozi i kros-meč rezultati</i>
<i>L7</i>	<i>Pregled, uzorak i selekcija</i>	<i>Statistike na nivou proizvođača podataka</i>
<i>L8</i>	<i>Spoljna usluga</i>	<i>Generalne povezane usluge</i>

Tabela 9: Semantički nivoi L0–L8 i njihova uloga.

Nivoi nisu međusobno isključive klase. Jedan pojam može biti dozvoljen u više konteksta, a deklaracija `level` važi u svom strukturnom opsegu dok je naredna deklaracija ne promeni. To se vidi u definiciji vremena:

```

<time>:
  level: L1, L2, L5, L6, L7, L8
  [mjd]:
  [datetime]:

level: L0
[first_mjd]:

```

```
[last_mjd]:
```

Putanje `time.mjd` i `time.datetime` opisuju vreme pojedinačnog ili kontekstualnog zapisa, dok `time.first_mjd` i `time.last_mjd` opisuju vremenski raspon na nivou objektnog sažetka. Oznake L0 i L1 ne ulaze u putanju; one se koriste kada druga struktura bira odgovarajući deo kontejnera.

`ontology.yaml` definiše očekivano ponašanje ovih projekcija, ali aktuelni leksički indeks i model semantičkih putanja sprovode uži skup strukturnih provera. Potpuno izračunavanje efektivnih nivoa i doslednosti njihovog prenošenja kroz sve reference još nije realizovano. Nivoi su zato formalni deo specifikacije, ali nisu tvrdnja da je svaka dozvoljena projekcija već potpuno automatski proverena ili dostupna kod svakog pružaoca podataka.

8.3.7 Kompozicija

Direktiva `$ref` uključuje postojeći kontejner u novu strukturu. Kod običnog kontejnera njegovo ime ostaje u putanji, dok oznaka nivoa u zagradi bira deo definicije namenjen datom kontekstu. Skraćeni iseći zapisa `summary` i `detection` pokazuju ovu razliku:

```
<summary>:
  $ref: <time>(L0)
  $ref: <identity>(L0)
  $ref: <position>(L0)
  $ref: <photometry>(L0)
```

```
<detection>:
  $ref: <position>(L1)
  $ref: <time>(L1)
  $ref: <identity>(L1)
  $ref: <quality>(L1)
  $ref: <provenance>
  $ref: <photometry>(L1)
```

Ovako iste osnovne definicije dobijaju različitu ulogu u dva zapisa:

```
summary.time.first_mjd
summary.position.ra
summary.photometry.g.psf.flux.mean
detection.time.mjd
```

`detection.position.ra`

`detection.photometry.g.psf.flux`

Selekcija nivoa primenjuje se do terminalnih polja, dok se kontejneri potrebni da se do njih stigne zadržavaju kao strukturni okvir. Zato izbor L0 agregata `mean` ne uklanja segmente `photometry`, oznaku filtra i `psf`. Referenca je istovremeno permisivna: ona određuje šta zapis može da sadrži, ali ne zahteva da svaki fizički odgovor popuni sva dozvoljena polja.

8.3.8 Korišćenje unutrašnjih projekcija

Strukture čije ime počinje donjom crtom služe kao gradivni elementi i ne mogu se samostalno materijalizovati kao javne putanje. Kada se takva struktura uključi pomoću `$ref`, prenosi se njen sadržaj, ali ne i njeno pomoćno ime. Klasifikacioni model koristi ovu mogućnost da isti skup polja primeni na najbolji rezultat i na pojedinačne izlaze klasifikatora:

```
<_classification_core>:
```

```
  [class]:
```

```
  [probability]:
```

```
    type: float
```

```
    range: [0, 1]
```

```
  [method]:
```

```
<classification>:
```

```
  <best>:
```

```
    $ref: <_classification_core>
```

```
  <assessment>:
```

```
    <{output}>:
```

```
      $ref: <_classification_core>
```

`classification.best.class`

`classification.best.probability`

`classification.assessment.rubin_classifier.class`

`classification.assessment.rubin_classifier.method`

Naziv `_classification_core` ne pojavljuje se u dobijenim putanjama. Projekcija samo centralizuje zajedničku strukturu, čime se smanjuje mogućnost da iste veličine u različitim delovima modela vremenom dobiju neusklađene nazive ili ograničenja (Gruber, 1993).

Unutrašnja projekcija ne mora uvek biti zasebno deklarirana. Postojeći javni kontejner može poslužiti kao izvor strukture, dok donja crta u referenci označava da se preuzima samo njegov sadržaj, bez imena samog kontejnera. To se koristi pri definisanju forsirane fotometrije:

```
<photometry>:
  <{filter}>:
    <psf>:
      [flux]:
        [error]:

<forced_photometry>:
  $ref: <_photometry>
```

```
forced_photometry.g.psf.flux
forced_photometry.g.psf.flux.error
```

Referenca `$ref: <_photometry>` preuzima polja i podkontekste kontejnera `photometry`, ali ne uvodi segment `photometry` u rezultujuću putanju. Zbog toga nastaje putanja `forced_photometry.g.psf.flux`, a ne `forced_photometry.photometry.g.psf.flux`. Ovim smo omogućili da izmerena i forsirana fotometrija koriste istu unutrašnju strukturu, jedinice i ograničenja, ali ostaju razdvojene svojim neposrednim semantičkim kontekstom. Time se izbegavaju prepisivanje definicija i naknadno razilaženje dva strukturno srodna dela modela.

8.3.9 Kardinalnost

Jednostruka referenca `$ref` i ponovljena referenca `$refs` razlikuju kardinalnost. U aktuelnoj ontologiji `$refs` se koristi u definiciji `portfolio` da označi da Portfolio može sadržati nula ili više instanci određene porodice zapisa:

```
<portfolio>:
  $refs: <summary>@{producer}:{channel}
  $refs: <detection>@{producer}:{channel}
  $refs: <crossmatch>@{producer}:{channel}
  $refs: <lightcurve>@{producer}:{channel}
  $refs: <classification>@{producer}:{channel}
```

Naziv zapisa ostaje u jednini, a indeks konkretne instance ne postaje deo javne semantičke putanje.

Više zapisa može zato imati istu putanju `detection@ztf:fink`; u izvršnom modelu razlikuju ih interni identifikatori i položaj u kolekciji.

Druga vrsta višestrukosti nastaje kada je vrednost jednog polja ili podkonteksta niz. Za nju se u aktuelnom modelu koristi `type: array`, kao kod tačaka svetlosne krive:

```
<lightcurve>:
  <points>:
    type: array
    items:
      $ref: <_lightcurve_point>
```

U prvom slučaju Portfolio sadrži ponovljene zapise, dok je u drugom slučaju `points` kolekcija unutar jednog zapisa `lightcurve`. Ova razlika opisuje aktuelnu ontologiju i izvršnu reprezentaciju prototipa; ne predstavlja opšte ograničenje svih mogućih modela ponovljenih struktura.

Svetlosna kriva objedinjuje prethodno uvedene mehanizme. Zajednički podaci o vremenu, identitetu, kvalitetu i poreklu najpre su izdvojeni u projekciju `_time_series_point`. Posebne projekcije zatim dodaju običnu ili forsiranu fotometriju:

```
<_time_series_point>:
  $ref: <time>(L1, L4)
  $ref: <quality>(L1)
  $ref: <identity>(L1)
  $ref: <provenance>

<_lightcurve_point>:
  $ref: <_time_series_point>
  $ref: <photometry>(L1)

<_forced_photometry_point>:
  $ref: <_time_series_point>
  $ref: <forced_photometry>(L1)
```

Zapis `lightcurve` koristi te projekcije kao tipove elemenata dve uređene kolekcije:

```
<lightcurve>:
  level: L4
  $ref: <provenance>
```

```

<points>:
  type: array
  items:
    $ref: <_lightcurve_point>
<forced_photometry_points>:
  type: array
  items:
    $ref: <_forced_photometry_point>

```

U aktuelnoj implementaciji normalizacija najpre proizvodi redno orijentisane fragmente, a zatim ih objedinjuje u jednu generisanu strukturu. Rezultat je jedan zapis `lightcurve`, čija polja sadrže uređene kolekcije detekcionih, forsiranih i drugih izvedenih tačaka. Svaka tačka može zadržati sopstveno vreme, fotometriju, identifikatore, pokazatelje kvaliteta i provenijencijske vrednosti, dok interni identitet semantičkog zapisa pripada svetlosnoj krivoj kao celini.

8.3.10 Semantičke veze

Hijerarhijske putanje opisuju sadržaj jednog zapisa. Odnos između dva zapisa ne dodaje se toj putanji, već se predstavlja u posebnoj ravni veza. Osnovni nosač veze - `--edge--` - definiše zajednička polja koja pripadaju samoj tvrdnji o odnosu:

```

--edge--:
  abstract: true
  level: L6
  [score]:
    type: float
    range: [0, 1]
  [basis]:
  $ref: <provenance>

```

Polja `score`, `basis` i `provenance` ovde ne opisuju ni polazni ni ciljni zapis, već vezu između njih. Struktura zapisa pomoću `$edges` navodi u kojim vezama može da učestvuje. Na primer, `detection` dozvoljava veze prema drugim detekcionim zapisima:

```

<detection>:
  $edges:
    --edge-- <detection>@{producer}:{channel}

```

```
detection@vidojevica --followup_of--> detection@lsst:fink
detection@lsst:fink --association-- detection@lvk:gcx
```

Strelica u nazivu označava usmerenu vezu, kao kod `--followup_of-->`, dok je `--association--` neusmerena. U izvršnom modelu konkretna veza predstavljena je objektom klase `SemanticEdge`, sa identifikatorima krajnjih zapisa i sopstvenim poljima.

Tek nakon definisanja ravni veza uvodi se direktiva `$is`. Njome se konkretna vrsta veze proglašava posebnom vrstom apstraktnog nosača `--edge--` i preuzima njegova zajednička polja:

```
--derived_from-->:
  $is: --edge--
  [method]:

--followup_of-->:
  $is: --edge--
  [request_id]:
  [program_id]:
```

Veza `--derived_from-->` zato može da nosi zajednička polja `score`, `basis` i `provenance`, kao i sopstveno polje `method`. Direktiva `$is` nije zamena za `$ref`: `$ref` sastavlja sadržaj imeničkog dela modela, dok `$is` nasleđuje osobine vrste veze i ne dodaje segmente tačkastoj putanji zapisa.

Kontejneri, zapisi, kvalifikatori, nivoi, reference, kolekcije i veze zajedno čine semantički prostor iz kojeg se grade konkretne instance. Sledeći odeljak razmatra kako se ti zapisi okupljaju oko identiteta astronomskog objekta u strukturi `Portfolio`.

8.4 Portfolio

`Portfolio` je osnovna semantička jedinica koja nosi identitet astronomskog objekta. On objedinjuje semantičke zapise, veze i pripadajuće podatke o poreklu, ali ne briše razliku između njihovih izvora. Kada više brokera dostavi podatke o istom paru (`origin`, `object_id`), sistem ih harmonizuje u jedan `Portfolio`. To je tehničko objedinjavanje zapisa koji dele identitet za razliku od naučne operacije *Match* kojom se utvrđuje da različiti identiteti verovatno predstavljaju isti fizički objekat na osnovu probabilističkih metoda, a koje su utemeljene u naučnoj literaturi (Williams et al., 2009; Graham et al., 2012).

Unutar jednog `Portfolio` objekta može postojati više detekcija, klasifikacija, kataloških

povezivanja ili drugih zapisa iste vrste. Kardinalnost se čuva na nivou izvršnih instanci i njihovih identifikatora, dok semantička putanja ostaje stabilna. Takva podela omogućava da se podaci akumuliraju kroz tok rada bez gubitka pojedinačnih tvrdnji i bez unošenja tehničkih indeksa u ontologiju.

```
<portfolio>:
  $refs: <summary>@{producer}:{channel}
  $refs: <detection>@{producer}:{channel}
  $refs: <crossmatch>@{producer}:{channel}
  $refs: <lightcurve>@{producer}:{channel}
  $refs: <color_magnitude>@{producer}:{channel}
  $refs: <color_color>@{producer}:{channel}
  $refs: <spectrum>@{producer}:{channel}
  $refs: <data_product>@{producer}:{channel}
  $refs: <classification>@{producer}:{channel}
  $refs: <survey>@{producer}:{channel}
  $edges:
    --association-- <portfolio>
    --spatially_near-- <portfolio>
    --possible_counterpart-- <portfolio>
    --host_of--> <portfolio>
```

Hijerarhijske putanje opisuju sadržaj koji pripada zapisu. Odnosi između zapisa predstavljeni su u posebnoj ravni semantičkih veza. Veza može, na primer, označiti asocijaciju, izvođenje jednog rezultata iz drugog ili odnos naknadnog posmatranja prema prethodnom događaju. Razdvajanje ravni podataka i ravni veza sprečava da se relacije prikivaju kao obična ugnežđena polja i omogućava da i same veze nose poreklo i kontekst tvrdnje.

Aktuelni izvršni model zato eksplicitno razlikuje SemanticRecord i SemanticEdge. Validator proverava poznate vrste zapisa, sintaksu relativnih putanja, dozvoljene simbole veza i postojanje njihovih krajnjih tačaka. Ovaj skup proverava već čuva osnovne invarijante strukture, iako potpuna interpretacija svakog polja u odnosu na sve nivoe ontologije ostaje deo buduće formalizacije.

8.5 Normalizacija

Ontologija svesno ne sadrži fizičke putanje brokerskih odgovora, nazive parametara krajnjih tačaka, transformacionu logiku niti način transporta. Ona određuje značenje i dozvoljenu strukturu.

Posebna deklarativna mapiranja (u datotekama *endpoints.yaml* za svaki par proizvođač-broker) opisuju gde se odgovarajuća vrednost nalazi u konkretnom REST odgovoru ili objektu Python klijenta, koje pomoćne transformacije su potrebne i koji se semantički zapis konstruiše. Time su fizičke promene izolovane od kanonskog modela.

Generički transformacioni sloj čita ova pravila i obavlja prevođenje reprezentacija (prema datotekama *mappings.yaml* za svaki par proizvođač-broker). Njegov konceptualni zadatak nije vezan ni za konkretan broker ni za jedan transportni protokol: ulazna vrednost se pronalazi prema deklarisanom ugovoru, po potrebi transformiše, a zatim upisuje na odgovarajuću semantičku putanju. Normalizacija iz tako dobijenih vrednosti konstruiše zapise i dodaje ih Portfolio objektu. Iz primera mapiranja kanonskih semantičkih putanja na konkretne vrednosti brokerskih odgovora naslućuje se složenost elemenata domena koje semantički model mora da opiše:

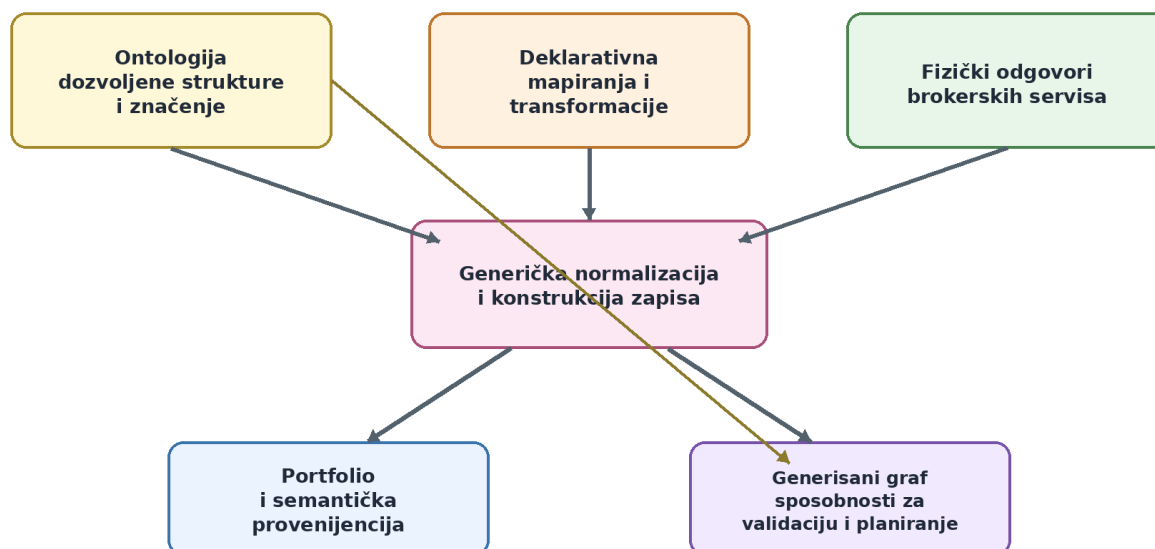
```
lightcurve@lsst:alerce.points.photometry.{filter}.psf.flags.edge:
```

```
- - query_lightcurve.detections#psfFlux_flag_edge
```

```
detection@lsst:alerce.image_metrics.pixel_flags.injected_center:
```

```
- - query_detections#pixelFlags_injectedCenter
```

Ovakvih deklarativnih mapiranja je popisano na stotine, po brokeru, po izvoru podataka. Iste deklaracije doprinose i generisanju grafa softverskih mogućnosti. Taj graf objedinjuje znanje o dostupnim krajnjim tačkama, vrstama rezultata, zahtevima ulaznih parametara, mapiranjima, transformacijama i semantičkom materijalu koji servis može da proizvede. Validacija i optimizator zbog toga ne proveravaju samo da li neka operacija postoji, već da li raspoloživi servis može da prihvati potrebne argumente i proizvede materijal koji je potreban narednim koracima. Otuda dodavanje novog servisa zahteva deklarisanje njegovog API ugovora i semantičkog mapiranja, ali ne i ugradnju posebne proceduralne grane u jezgro sistema. Sistem je tehnološki-implementaciono agnostičan u tom smislu.

Uloga semantičke specifikacije u obradi podataka

Slika 9: Uloga ontologije, deklarativnih mapiranja i fizičkih odgovora u normalizaciji podataka, konstrukciji Portfolio sadržaja i generisanju grafa sposobnosti za validaciju i planiranje.

8.6 Provenijencija

U Alertissimu se razlikuju dva nivoa provenijencije. Semantička provenijencija pripada modelu podataka i odgovara prvenstveno na pitanje odakle potiče određeni normalizovani podatak. Ona čuva neposredni izvor semantičkog zapisa ili veze i omogućava da se nakon harmonizacije i dalje razlikuju tvrdnje različitih proizvođača i kanala dostave.

Tehnička, odnosno provenijencija izvršenja, odvojena je od semantičkog modela. Ona odgovara na pitanje kako je rezultat proizveden u konkretnom izvršenju toka rada: kojim fizičkim korakom, preko kog provajdera podataka i krajnje tačke izvršenja, sa kojim parametrima i u kakvoj vezi sa prethodnim koracima. Zbog toga dva izvršenja mogu proizvesti semantički sličan materijal, ali imati različitu tehničku istoriju.

Ova podela je praktična i namerna. Semantička provenijencija ostaje dovoljno blizu podatku da podrži tumačenje rezultata, dok se izvršni audit može razvijati bez opterećivanja kanonskog modela

detaljima transporta i izvršenja. Alertissimo pri tome nije zamišljen kao potpuni sistem distribuiranog praćenja, niti semantička provenijencija u složenim agregatima nužno čuva izvornu poziciju svakog elementa. Za šire standardizovano modelovanje porekla relevantan je IVOA Provenance Data Model, koji obuhvata entitete, aktivnosti, agente i njihove odnose (Moreau & Missier, 2013; Servillat et al., 2020).

8.7 Standardi

Alertissimo ontologiju ne treba predstavljati kao zamenu za standarde Međunarodne Unije virtuelnih opservatorija. VO-DML pruža standardizovan metamodel za opis IVOA modela podataka i nezavisan je od konkretne fizičke serijalizacije (Lemson et al., 2018). MIVOT povezuje elemente VOTable-a sa instancama modela izraženih u VO-DML-u (Michel et al., 2023). ObsCore definiše minimalni skup metapodataka za otkrivanje opservacionih skupova kroz TAP servise (Louys et al., 2017), dok VOEvent uređuje razmenu informacija o tranzijentnim događajima (Seaman et al., 2011). UCD oznake i IVOA kontrolisani rečnici (Derriere et al., 2005, Cecconi et al., 2024) mogu dopuniti pojedinačne fizičke veličine i termine, ali ne prenose celokupan strukturni kontekst hijerarhijske putanje.

Navedeni standardi rešavaju susedne, ali ne identične probleme. Alertissimo je neposredno usmeren na operativno prevođenje heterogenih REST/JSON i Python reprezentacija brokerskih servisa u zajedničku strukturu, uz transformacije, kvalifikovane izvore, Portfolio harmonizaciju i vezu sa planiranjem izvršenja. VO-DML i MIVOT nude jaču standardizaciju i institucionalnu interoperabilnost, ali bi za brokersku federaciju zahtevali dodatne ugovore za fizička mapiranja, transformacije a naročito za sprovođenje same orkestracije.

Od opštih standarda i metoda za modelovanje složenih sistema, OWL 2 pruža formalno definisanu semantiku za opis klasa, svojstava, relacija, ekvivalencija i ograničenja, kao i mehanizme automatskog zaključivanja nad tako predstavljenim znanjem. Njime se, na primer, može iskazati da je određen pojam specijalizacija drugog pojma, da su dve klase međusobno disjunktne ili da postojanje jedne relacije implicira dodatne semantičke veze. OWL 2 je zato naročito pogodan za formalizaciju znanja i proveru logičke konzistentnosti ontologije. Međutim, on sam po sebi ne određuje kako će se podatak pronaći u konkretnom REST odgovoru ili objektu Python klijenta, kako će fizički naziv polja biti preveden u kanonsku semantičku putanju niti kako će više odgovora biti objedinjeno tokom izvršenja toka rada (W3C OWL Working Group, 2012).

SHACL ima užu, ali operativno veoma značajnu ulogu. Njime se nad RDF grafovima mogu definisati i proveravati očekivani tipovi vrednosti, kardinalnosti, dozvoljeni opsezi, obavezna svojstva i složenija ograničenja nad putanjama. SHACL bi, na primer, mogao da proveri da li koordinata ima numeričku vrednost, da li je verovatnoća u intervalu od nula do jedan ili da li određeni tip zapisa sadrži propisani identifikator. Ipak, takva validacija pretpostavlja da je ulazni materijal već preveden u odgovarajući RDF graf i povezan sa poznatim semantičkim pojmovima. SHACL ne rešava samostalno pitanje kako iz heterogenih brokerskih odgovora nastaje taj graf, kako se interpretiraju ugnežđeni nizovi i objekti niti kako se razlikuju detekcije, nedetekcije, forsirana fotometrija i agregatni podaci kada ih različiti servisi vraćaju u različitim fizičkim strukturama (W3C RDF Data Shapes Working Group, 2017).

LinkML zauzima praktičniju poziciju između formalnog semantičkog modelovanja i razvoja softvera. On omogućava da se klase, slotovi, tipovi, kardinalnosti i kontrolisani rečnici opišu u relativno čitljivom šematskom obliku, iz kojeg se zatim mogu generisati različite reprezentacije, validatori i programski artefakti. Takav pristup može biti koristan za dokumentovanje kanonskih struktura Alertissima, razmenu modela sa drugim sistemima ili buduće generisanje delova implementacije. Međutim, ni LinkML ne definiše automatski operativnu vezu između naučnog značenja i konkretnih parametara krajnje tačke izvršenja, lokacije vrednosti u odgovoru, pravila transformacije i akumulacije podataka kroz višestepeni tok rada. Te veze bi i dalje morale biti opisane dodatnim mapiranjima i izvršnim pravilima.

Zbog toga navedeni pristupi nisu suprotstavljeni semantičkom modelu Alertissima, već predstavljaju moguće komplementarne slojeve i ciljne reprezentacije. OWL 2 može doprineti strožoj formalizaciji i zaključivanju, SHACL validaciji grafovski predstavljenih podataka, a LinkML razmeni šema i generisanju pratećih artefakata. Specifičan doprinos Alertissima nalazi se u operativnom povezivanju tih značenja sa heterogenim servisima: deklarativnim opisivanjem fizičkih reprezentacija, njihovim prevođenjem u zajedničke semantičke strukture, očuvanjem provenijencije i korišćenjem normalizovanog materijala u daljoj orkestraciji. Opšti standardi, dakle, mogu formalizovati ili validirati pojedine delove rešenja, ali nijedan od njih samostalno ne zamenjuje sloj mapiranja, transformacije i normalizacije potreban za izvršavanje interoperabilnih tokova rada u Alertissimu.

Pristup	Primarna snaga	Odnos prema Alertissimu
<i>VO-DML</i>	<i>Standardizovano konceptualno modelovanje</i>	<i>Moguća formalna projekcija kanonskog modela</i>
<i>MIVOT</i>	<i>VOTable–VO-DML mapiranje</i>	<i>Komplementarno mapiranjima nad sadržajem brokerskih odgovora</i>
<i>UCD i IVOA rečnici</i>	<i>Semantika fizičkih veličina i kontrolisani termini</i>	<i>Moguće spoljne oznake pojedinačnih pojmova</i>
<i>VOEvent</i>	<i>Razmena događaja</i>	<i>Komplementaran format komunikacije</i>
<i>ObsCore</i>	<i>Otkrivanje opservacionih skupova</i>	<i>Komplementaran model za arhivsku pretragu</i>
<i>IVOA Provenance</i>	<i>Standardizovano poreklo entiteta i aktivnosti</i>	<i>Širi model od praktične interne provenijencije</i>
<i>OWL i SHACL</i>	<i>Zaključivanje i validacija RDF grafova</i>	<i>Mogući izvozni ili dodatni formalni sloj</i>
<i>LinkML</i>	<i>Šeme, generatori i interoperabilnost</i>	<i>Moguća projekcija bez zamene unutrašnjeg modela</i>

Tabela 10: Odnos Alertissimo semantičkog modela prema izabranim standardima i tehnologijama.

Komplementarni odnos prema standardima bi mogao biti najadekvatniji. Unutrašnji model može ostati prilagođen heterogenim brokerskim odgovorima, dok se na njegovim granicama uvode veze prema standardnim identifikatorima, VOTable anotacijama, VO-DML modelima ili RDF/SHACL projekcijama. Time se čuva operativna jednostavnost, a istovremeno otvara put ka široj interoperabilnosti i upotrebi postojećih VO alata.

8.8 Formalizacija

Aktuelna ontologija predstavlja uređenu, čitljivu specifikaciju nalik YAML-u, ali se ne može tumačiti kao običan YAML dokument. Redosled deklaracija, njihovo ponavljanje i opseg važenja imaju semantičko značenje, pa jednostavno učitavanje sadržaja u strukturu tipa rečnika nije dovoljno za njeno potpuno tumačenje. Postojeći `SemanticModelIndex` namerno ima ograničenu ulogu: leksički evidentira osnovne kontejnere, polja i vrste veza. `SemanticPathModel` nad tim opisom omogućava strukturnu proveru semantičkih putanja i vrsta zapisa, dok se pri obradi podataka dodatno proveravaju poznate vrste zapisa, ispravnost relativnih putanja i dozvoljene veze u strukturi `Portfolio`.

Postojeće provere omogućavaju da se ontologija neposredno koristi pri proveri deklarativnih mapiranja i konstrukciji normalizovanih semantičkih zapisa, ali ne predstavljaju potpunu proveru ispravnosti same ontološke specifikacije. Trenutna implementacija, između ostalog, ne obuhvata sistematsku proveru postojanja ciljeva svih referenci, otkrivanje i ocenu cikličnih referenci prema pravilima kompozicije, niti potpuno tumačenje opsega i prenošenja semantičkih nivoa L0–L8. Za takvu proveru potreban je potpuniji mehanizam za čitanje uređene specifikacije, koji bi sačuvao značenje ponovljenih pravila i njihovog opsega, razrešio reference i izračunao efektivne semantičke nivoe. Formalno definisana pravila samog jezika ontologije zatim bi omogućila sistematsku proveru svojstava kompozicije, kardinalnosti, referenci i veza, kao i izradu izveštaja o uočenim nedoslednostima.

Drugi pravac predstavlja povezivanje kanonskih pojmova sa spoljnim identifikatorima. Gde postoje odgovarajući UCD, IVOA rečnik ili VO-DML koncept, veza može biti dodata kao metapodatak bez nametanja njihove fizičke serijalizacije unutrašnjem sistemu. Iz istog formalizovanog jezgra zatim je moguće generisati dokumentaciju ili projekcije ka VO-DML-u, LinkML-u i SHACL-u.

Treći pravac je proširenje pokrivenosti uz očuvanje granica odgovornosti. Ontologija treba da određuje značenje i dozvoljenu strukturu; mapiranje fizičko poreklo i konverziju; izvršni model instance, veze i neposrednu provenijenciju; graf sposobnosti izvršivu ponudu servisa; a orkestracija redosled i mesto obrade. Ovakva podela omogućava razvoj pojedinačnih delova bez pretvaranja semantičkog modela u dokumentaciju jednog API-ja ili jednog programskog jezika.

Hijerarhijski semantički model time predstavlja jedan od ključnih vezivnih elemenata arhitekture

Alertissima. On omogućava da heterogeni odgovori postanu uporediv i akumulativan materijal, da se planiranje zasniva na značenju rezultata umesto na nazivima tačaka izvršenja i da se poreklo podataka očuva kroz višestepeni tok. Njegova praktična i ontološka uloga nisu konkurentske: upravo njihovo povezivanje omogućava da formalizacija ostane izvršiva, a izvršenje semantički interpretabilno.

9 Sloj za orkestraciju i izvršenje

Sloj za orkestraciju i izvršenje rešava komplementaran problem semantičkoj harmonizaciji predstavljenoj u prethodnom poglavlju: kako naučnu nameru predstaviti kao proverljiv niz operacija, kako za svaku operaciju pronaći moguću fizičku realizaciju i kako kroz više koraka očuvati kandidate, akumulirani semantički materijal i podatke o konkretnim izvršenjima. Na ovaj problem možemo gledati i kao na kompoziciju funkcija koje transformišu i prenose stanje u smislu računskih monada (Moggi, 1991; Wadler, 1992). Centralna međuprezentacija ovog sloja jeste IR model radnog toka (engl. *intermediate representation*), predstavljen kao kompozitni, uređeni tok koraka (implementiranih u vidu Step objekata). Princip međuprezentacije, iako prvobitno definisan na kompajlerskoj infrastrukturi (Lattner & Adve, 2004) nalazi svoje konceptualno mesto u sistemima za orkestraciju i kontrolu tokova rada (Coleman et al., 2022; Wang et al., 2026).

Pri modelovanju sistema pošli smo od principa da IR predstavlja kanonski prikaz toka rada nakon što je korisnička namera formalizovana (Vujčić et al., 2026.). Ovaj skup koraka je potpuno nezavisan od ulaza u sistem. DSL je jedan ulazni jezik kojim se IR može izgraditi, a budući interfejsi prirodnog jezika, vizuelni editor ili neki četvrti interfejs mogu proizvesti isti rezultat. Kada je prevod završen, parser, tekst originalnog DSL iskaza i elementi korisničkog interfejsa nisu potrebni optimizatoru, sloju za povezivanje, izvršnom sloju ili normalizaciji. Ovi slojevi rade nad WorkflowIR-om i njegovim projekcijama.

Ova višeslojna podela razlikuje Alertissimo od konvencionalnih sistema koji radni tok neposredno predstavljaju pukim nizom poziva API funkcija ili krajnjih tačaka izvršenja. Arhitektura sistema uvodi razdvajanje odgovornosti za izvršavanje kroz pet nivoa apstrakcije:

- Korak toka rada (Step) formalno opisuje atomsku konceptualnu celinu zahteva;
- Plan izvršenja (EndpointPlan) definiše gde i na koji način ta apstraktna namera može biti fizički realizovana;
- Vezani poziv (BoundEndpointCall) enkapsulira konkretne fizičke parametre generisane za specifičnu izvršnu instancu;
- Rezultat izvršenja (ExecutionResult) donosi sirovi fizički odgovor infrastrukture zajedno

sa zabeležkom o tehničkoj provenijenciji;

- **Portfolio** predstavlja finalni, semantički-harmonizovani rezultat vraćen u domen naučnih podataka.

Arhitektonska separacija slojeva eliminiše zavisnost od načina imenovanja eksternih servisa.

Poklapanje naziva između API metode brokera i naučne operacije ne implicira nužno njihovu ekvivalentnost; ime tačke izvršenja poziva je tek fizička mrežna lokacija, ne i semantička definicija samog naučnog koraka (Vujčić, Srećković i Babarogić, 2026).

9.1 Granice odgovornosti orkestracionog sloja

Orkestracioni sloj povezuje semantičku nameru i izvršivu infrastrukturu, ali ne treba da preuzme odgovornosti susednih slojeva. Semantički model podataka određuje značenje zapisa i polja. Deklaracije provajdera opisuju njihove fizičke ugovore, ulazne parametre i izlazne reprezentacije. Generički transformacioni sloj obavlja konverzije kolekcija, tipova i složenih vrednosti. Izvršni (pod)sloj nezavisno poziva REST servis ili Python klijent. Normalizacija pretvara odgovor u `SemanticRecord`, `SemanticEdge` i `Portfolio`. Orkestracija koordinira ove komponente, određuje redosled, zavisnosti i mesto izvršenja, ali ne ugrađuje u centralnu logiku posebna pravila za svaki broker (Parnas, 1972; Shaw & Garlan, 1996; Martin, 2017).

Ova granica može se izraziti kroz nekoliko invarijanti. `WorkflowIR` ne sadrži URL, HTTP metod, naziv Python klase, fizički naziv parametra ili sirov odgovor. `EndpointPlan` ne menja značenje `Step`-a, već samo beleži izabranu fizičku mogućnost izvršenja. `Binder` ne tumači ponovo naučni uslov, već primenjuje već ustanovljenu vezu između kanonske uloge i fizičkog parametra. `Executor` ne odlučuje da li je poziv semantički pogodan, već izvršava validiran `EndpointSpec`. `Normalizer` ne zaključuje naučnu nameru iz naziva krajnje tačke izvršenja (engl. *endpoint*), već primenjuje deklarativno mapiranje pripadajućeg odgovora.

Posledica je da se jedna semantička operacija može realizovati jednim pozivom, većim brojem poziva ili bez poziva provajdera. Obrnuto, jedan fizički poziv može zadovoljiti više semantičkih potreba ili proizvesti više `Portfolio` objekata. Ove kardinalnosti nisu izuzeci već očekivani deo modela.

9.2 WorkflowIR kao kanonska reprezentacija toka

WorkflowIR je definisan kao ureden niz Step objekata, upotpunjen imenom i opisom samog toka. Redosled ovih objekata određuje sekvencu semantičkih operacija, prateći principe semantičkog modeliranja i validacije apstraktnih tokova koje su postavili Gil et al., 2007. Sa druge strane, sam WorkflowIR je potpuno lišen stanja izvršenja (engl. *stateless*). Ova konceptualna podela na nepromenljivu specifikaciju toka i dinamičku istoriju njegovog izvršavanja osigurava visoku sledivost i provenijenciju podataka prema modelu koji opisuju Ludäscher et al., 2006. Usled ovakve arhitekture, isti WorkflowIR objekat može biti validiran, optimizovan i mapiran na resurse neograničen broj puta, gde svaka pojedinačna instanca izvršenja rezultuje kreiranjem zasebnog WorkflowRun objekta – što predstavlja standardnu praksu u robusnim sistemima za raspoređivanje i orkestraciju kompleksnih procesa (Deelman et al., 2009).

```
WorkflowIR(
  name="candidate enrichment",
  steps=[
    SemanticSearchStep(
      semantic_type="summary",
      sources=[Source(broker="alerce", origin="lsst")]
    ),
    FilterStep(predicate=selection_predicate),
    GetLightcurveStep(
      sources=[Source(broker="fink", origin="lsst")]
    )
  ]
)
```

Primer ne navodi API/endpoint za pretragu, fizičko ime identifikatora niti format zahteva. Prvi korak otkriva objedinjene semantičke zapise, drugi lokalno sužava skup kandidata, a treći pribavlja svetlosne krive za semantičke zapise koji su zadovoljili uslove filtriranja. Optimizator (Planner) tek naknadno povezuje korak sa odgovarajućim brokerskim pozivima, te određuje kako će identifikatori iz prvog ili drugog koraka postati ulaz trećeg.

WorkflowIR koristi diskriminisanu uniju koraka: polje prima konkretne tipove Step-a, a modeli odbijaju nepoznata polja. Time IR nije proizvoljan rečnik, već proverljiv ugovor između svih ulaznih interfejsa i orkestracije. Važno je i da Step označava jednu semantičku pojavu u toku. Čak i kada optimizator dokaže da dva Step-a mogu koristiti isto fizičko izvršenje, oba ostaju prisutna u

WorkflowIR-u i zadržavaju svoje mesto i značenje.

9.3 Zajednički elementi IR koraka

Svi Step modeli mogu sadržati listu Source ograničenja. Source ne bira API/endpoint, već sužava prostor dozvoljenih realizacija prema konkretnom brokeru, proizvođaču podataka ili prema te dve uparene osobine. Prazna lista znači da optimizator može razmatrati čitav registrovani prostor mogućnosti.

```
Source(broker="fink", origin="lsst")
Source(broker="lasair")
Source(origin="ztf")
```

Za operacije nad unapred poznatim entitetima koristi se TargetSelector. On sadrži nepraznu listu jedinstvenih identifikatora i vrstu identifikovane celine: object(s) i alert(s). TargetSelector je vrednost unutar odgovarajućeg Step-a, ne zaseban korak toka.

```
TargetSelector(
    ids=["A", "B"],
    kind="object"
)
```

Vremenska ograničenja predstavljena su objektom TimeContext. On može opisati apsolutni početak i kraj, relativni prozor tj. okvir, referentni trenutak, pomeraj ili uzorkovanje, bez prevođenja tih pojmova u parametre pojedinačnog provajdera. Tek binder i deklaracije fizičkog ugovora određuju da li će, na primer, vremenski okvir postati startdate i stopdate, interval u upitu ili rezidualni lokalni uslov.

```
TimeContext(
    start_time="2026-01-01T00:00:00Z",
    end_time="2026-01-31T23:59:59Z"
)
```

9.4 Porodice semantičkih Step objekata

IR rečnik namerno razlikuje otkrivanje kandidata, pribavljanje već postojećih podataka, lokalno izvođenje novih proizvoda, naučno povezivanje, analizu, potvrđivanje, praćenje i spoljne akcije. Ove razlike čine osnovu sistema, i zasnivaju se na principima predstavljanim u poglavlju 2 te ostaju

semantički relevantne čak i kada bi neki budući provajder više operacija mogao da realizuje jednim pozivom. Tabela 11 sažima osnovne porodice koraka tokova rada i istovremeno razdvaja formalno prisustvo u IR-u od stepena njihove izvršne podrške.

Porodica IR koraka	Semantička uloga	Stanje u aktuelnoj implementaciji
<i>LookupStep</i>	<i>Razrešavanje već poznatog identifikatora</i>	<i>Fizička podrška zavisi od provajdera i porekla</i>
<i>SearchStep</i>	<i>Otkrivanje kandidata prema semantičkim uslovima</i>	<i>SemanticSearch, ConeSearch i SQL-varijanta postoje; fizička podrška zavisi od registra</i>
<i>FilterStep</i>	<i>Lokalno sužavanje već formiranog skupa kandidata</i>	<i>Izvršava se nad normalizovanim Portfolio materijalom, bez poziva provajdera</i>
<i>GetStep</i>	<i>Preuzimanje već postojećeg podatka, proizvoda ili tvrdnje</i>	<i>Planiranje, povezivanje parametara i izvršenje rade za registrovane i kompatibilne tačke izvršenja</i>
<i>DeriveStep</i>	<i>Lokalna konstrukcija novog semantičkog proizvoda</i>	<i>ColorMagnitude i ColorColor su izvršivi; lokalna konstrukcija Lightcurve još nije realizovana</i>
<i>MatchStep</i>	<i>Naučno povezivanje entiteta ili zapisa</i>	<i>Implementirana je prva poziciona varijanta; šire geometrijske i vremenske korelacije su u planu</i>
<i>AnalyzeStep</i>	<i>Lokalna analiza, klasifikacija,</i>	<i>IR rečnik postoji; opšti izvršni</i>

Porodica IR koraka	Semantička uloga	Stanje u aktuелnoj implementaciji
	<i>agregacija ili poređenje</i>	<i>mehanizam za sve podtipove još nije integrisan</i>
<i>ConfirmStep</i>	<i>Potvrđivanje kandidata kvorumom nezavisnih brokera</i>	<i>Hibridno izvršenje je implementirano: udaljeno pribavljanje dokaza i lokalna provera kvoruma</i>
<i>MonitorStep</i>	<i>Dugotrajno praćenje semantičkog toka</i>	<i>Semantički definisan; kontinuirano praćenje nije implementirano</i>
<i>ActionStep</i>	<i>Spoljni efekat: zahtev za praćenje, obaveštenje ili izvoz</i>	<i>IR podtipovi postoje; okidači akcija nisu implementirani</i>

Tabela 11: Porodice semantičkih IR koraka i njihov status u aktuелnoj implementaciji.

9.4.1 Razrešavanje identifikatora i otkrivanje kandidata

LookupStep polazi od već poznatog identifikatora. On je odvojen od SearchStep-a jer pretražuje konkretnu instancu ili instance objekta ili alerta.

```
LookupStep(
    target=TargetSelector(
        ids=["ZTF18abbuksn"],
        kind="object"
    ),
    sources=[Source(origin="ztf")]
)
```

SearchStep predstavlja otkrivanje kandidata. Njegovi konkretni podtipovi su SemanticSearchStep, ConeSearchStep i SqlQueryStep. SemanticSearchStep omogućava selekciju na osnovu atributa iz celog atributskog prostora, nezavisno od toga da li je pojedini predikatski atribut podržan kao parametar u pozivu brokera. ConeSearchStep eksplicitno predstavlja geometriju konusa kroz rektascenziju, deklinaciju i ugaoni poluprečnik. SearchSelection može dodati izbor poslednjih N kandidata, pri čemu latest ima globalno

semantičko značenje i ne mora biti identičan lokalnom limit parametru svakog provajdera.

```
ConeSearchStep(
    semantic_type="summary",
    ra=210.8021,
    dec=54.3482,
    radius=3.0,
    sources=[Source(broker="fink", origin="ztf)],
    selection=SearchSelection(latest=100)
)
```

SqlQueryStep je eksplicitna mogućnost za servis koji oglašava SQL-sličnu sposobnost. Ovaj podtip treba posmatrati kao kontrolisanu mogućnost zadavanja naprednih ili još neformalizovanih upita, a ne kao zamenu za kanonski SearchStep model. Služi najpre da podrži korisnike koji ne žele da se odreknu izvornog načina i ekspresivnosti upita prema brokeru koji omogućava takve pozive (npr Lasair).

9.4.2 Pretraga i lokalno filtriranje

SemanticSearchStep i FilterStep mogu koristiti isti jezik predikata, ali pripadaju različitim fazama životnog ciklusa. SearchStep definiše uslov otkrivanja kandidata i može biti delimično ili potpuno prenet udaljenom servisu. FilterStep deluje nad podacima koji su već pribavljeni, normalizovani i prisutni u trenutnom semantičkom kontekstu. On zato nema EndpointPlan i ne proizvodi novu tehničku provenijenciju poziva.

```
SemanticSearchStep(
    semantic_type="summary",
    predicate=brightness_predicate
)
```

```
FilterStep(
    predicate=brightness_predicate
)
```

Iako su predikati jednaki, prvi iskaz znači „pronađi kandidate koji zadovoljavaju uslov”, a drugi „među već pribavljenim kandidatima zadrži one koji zadovoljavaju uslov”. Ovo razdvajanje omogućava optimizatoru da menja mesto izvršenja bez menjanja značenja.

9.4.3 Pribavljanje postojećeg materijala i lokalno izvođenje

GetStep označava pribavljanje već postojećeg podatka, proizvoda ili tvrdnje. Njegovi podtipovi obuhvataju svetlosnu krivu, kros-meč, isečak slike, forsiranu fotometriju, klasifikaciju, spektar i opšti proizvod podataka. Get operacija ne zahteva nužno da servis generiše novo posmatranje ili proizvod; ona traži raspoloživi materijal.

```
GetLightcurveStep(  
    target=TargetSelector(ids=["A", "B"], kind="object"),  
    bands=["g", "r"],  
    sources=[Source(broker="fink", origin="lsst")]  
)
```

DeriveStep ima drugačiju semantiku: on lokalno konstruiše novi semantički proizvod iz već normalizovanog Portfolio materijala. Zato GetLightcurveStep i LightcurveStep nisu sinonimi. Prvi pribavlja svetlosnu krivu ili odgovarajuću istoriju koju provajder već izlaže, dok drugi treba da napravi lokalno obogaćenje - npr da konstruiše kompozitnu svetlosnu krivu iz svih dostupnih detekcija i fotometrijskih zapisa. Aktuelni izvršni sloj može da izračuna ColorMagnitudeStep i ColorColorStep derivacije, dok su pravila lokalne LightcurveStep konstrukcije još otvorena i ova operacija nije izvršno realizovana.

```
GetLightcurveStep(...) # udaljeno pribavljanje postojećeg materijala  
LightcurveStep(...) # lokalna konstrukcija novog proizvoda
```

GetLightcurveStep je dobar primer zašto Step ne sme da bude poistovećen sa krajnjom tačkom izvršenja. Fink/LSST razdvaja običnu istoriju *diaSource* merenja i forced fotometriju u krajnje tačke *sources* i *fp*. Optimizator (Planner) može jedan semantički GetLightcurveStep razložiti na obavezni sources plan i dopunski fp plan. Kod provajdera koji već vraća objedinjenu istoriju dovoljan je jedan fizički poziv. WorkflowIR u oba slučaja ostaje isti.

9.4.4 Povezivanje, ukrštanje i poređenje

Tri naizgled slična zahteva predstavljena su različitim Step modelima. GetCrossmatchStep pribavlja rezultat ukrštanja koji je već proizveo broker, katalog ili drugi servis. MatchStep lokalno utvrđuje povezanost dve astronomske celine ili dva zapisa prema pozicionom, vremenskom,

probabilističkom ili drugom kriterijumu. CompareStep meri slaganje ili razliku između već izabranih vrednosti i tvrdnji, ali ne uspostavlja identitet ili naučnu asocijaciju objekata (Budavári & Szalay, 2008).

```
GetCrossmatchStep(catalog="gaia")
```

```
MatchStep(
    method="position",
    params={
        "candidate_origins": ["ztf", "lsst"],
        "max_angular_separation_arcsec": 1.0
    }
)
```

```
CompareStep(
    comparison_target="classification.best.class",
    method="agreement"
)
```

Aktuelni lokalni Match izvršilac realizuje prvu precizno definisanu varijantu: poziciono povezivanje kandidata iz eksplicitno navedenih porekla. Prihvaćeni parovi dobijaju semantičke - - spatially_near - - veze, a samo objekti koji učestvuju u najmanje jednoj prihvaćenoj relaciji ostaju u rezultatu tog Step-a. Širi ugovori za vremensko povezivanje i za povezivanje sa odgovarajućim objektima u drugim izvorima ostavljeni su za naredne iteracije.

9.4.5 Analiza, potvrđivanje, monitoring i akcije

AnalyzeStep je zajednička osnova za lokalnu analitiku. MethodAnalysisStep omogućava imenovani otvoreni algoritam, dok ClassifyStep, AggregateStep, CompareStep i UtilityScoreStep eksplicitno razlikuju klasifikaciju, agregaciju, poređenje i procenu korisnosti kandidata. Ovi modeli formalizuju budući izvršni ugovor, ali opšti mehanizam za izvršavanje svih analitičkih podtipova još nije integrisan u optimizator i izvršni sloj. Zato njihovo prisustvo u IR-u ne treba pogrešno predstavljati kao potpunu izvršnu podršku.

ConfirmStep je hibridna operacija: najpre izvršava registrovane pozive, zatim normalizuje njihovu strukturu i lokalno primenjuje kvorum. Jedan broker daje najviše jedan glas za isti identitet, bez

obzira na broj zapisa koji je vratio. Predikat je opcioni: bez njega se potvrđuje postojanje identiteta, a sa njim svaki broker mora sopstvenim normalizovanim materijalom potvrditi istu semantičku tvrdnju.

```
ConfirmStep(  
    sources=[  
        Source(broker="fink", origin="ztf"),  
        Source(broker="alerce", origin="ztf"),  
        Source(broker="lasair", origin="ztf")  
    ],  
    required_agreement=2,  
    predicate=classification_predicate  
)
```

MonitorStep opisuje dugotrajno praćenje semantičkog toka, ali transporti pretplate, održavanje stanja, rekonekcija, kašnjenje i kontinuirano praćenje nisu implementirani. ActionStep i njegovi podtipovi FollowupRequestStep, NotifyStep i ExportStep opisuju spoljne efekte, ali njihovi okidači takođe nisu deo trenutnog sistema.

9.5 Semantički predikati u IR-u

Predikati su zasebna kanonska reprezentacija uslova i zajednički su za pretragu, filtriranje i potvrđivanje. Ne čuvaju tekst DSL izraza niti fizičke parametre provajdera. Osnovni operand je SemanticReference, sastavljen od porodice semantičkog zapisa, relativne putanje polja i opcionog proizvođača i kanala. Vrednost se predstavlja objektom PredicateLiteral.

```
ComparisonPredicate(  
    operator=">=",  
    left=SemanticReference(  
        semantic_type="classification",  
        field_path="best.probability",  
        producer="stamp_classifier_rubin_beta_20260421"  
    ),  
    right=PredicateLiteral(value=0.5)  
)
```

Složeniji izrazi grade stablo pomoću BooleanPredicate, NotPredicate i ExistsPredicate

čvorova. `BooleanPredicate` zahteva najmanje dva operanda, a operator može biti `and` ili `or`. `SemanticReference` koristi relativnu putanju bez znaka `@` unutar `field_path`, jer se kvalifikator proizvođača i kanala vodi odvojeno i pripada prvostepenom semantičkom zapisu.

```
BooleanPredicate(
    operator="and",
    operands=(
        ComparisonPredicate(...),
        ExistsPredicate(
            reference=SemanticReference(
                semantic_type="lightcurve",
                field_path="points"
            )
        )
    )
)
```

Struktura iz primera omogućava optimizatoru (`Planner`) da svaki segment uslova analizira potpuno nezavisno. Delovi uslova za koje postoji deklarativno dokazano ekvivalentno mapiranje ka parametrima krajnje tačke transformišu se u potisnute predikate (engl. *pushdown predicates*). Nasuprot njima, preostali uslovi zadržavaju status rezidualnih (engl. *residual*) i evaluiraju se naknadno, nad normalizovanim `Portfolio` zapisima u memoriji. Unutar generisanog plana (`EndpointPlan`), i potisnuti i rezidualni uslovi ostaju čisti semantički predikati koji definišu logiku filtera. S druge strane, konkretne fizičke vrednosti izvedene za dati izvršni poziv izoluju se isključivo unutar parametarske strukture. Ovakvo striktno razdvajanje semantičkog opisa filtera od njegove fizičke instancijacije u vremenu izvršavanja se oslanja na principe optimizacije i separacije faza koje su definisali Hellerstein i Stonebraker (1993).

9.6 Validacija mogućnosti i izvršni graf

Proces provere validnosti unutar sistema odvija se u dve posebne faze. Prva faza obuhvata formalnu validaciju pomoću Pydantic modela, kojom se verifikuje sama struktura i parametarski integritet IR-a: proveravaju se dozvoljene vrednosti, obavezna polja, ispravnost koordinatnih opsega rektascenzije i deklinacije kod konusne pretrage (engl. *cone search*), kao i jedinstvenost ciljnih identifikatora i druge strukturne invarijante. Druga faza, odnosno validacija softverskih mogućnosti ili sposobnosti (engl. *capability validation*), fokusira se na funkcionalnu izvodljivost: njome se

utvrđuje da li registrovani eksterni sistemi u datom trenutku raspolažu fizičkim resursima i funkcionalnostima koje mogu zadovoljiti zahteve definisane unutar Step objekta. Provera podudarnosti konceptualno se oslanja na koncepte semantičkog uparivanja sposobnosti veb servisa koje su postavili Paolucci et al., 2002.

CapabilityGraph se generiše iz deklaracija svih provajdera. On povezuje identitet krajnjih tačaka izvršenja API-ja, oglašene vrste operacija, zapise koje odgovor može proizvesti, mapirana semantička polja, transformacije i ugovore vezivanja parametara. Validacija graf samo čita. Ona ne bira krajnju tačku, ne prevodi argumente, ne izvršava poziv i ne spaja rezultate.

Rezultat validacije ima jedno od četiri stanja: *supported* znači da postoji dokazano kompatibilan kandidat; *unsupported* znači da zahtev nije moguće zadovoljiti postojećim deklaracijama; *not_applicable* se koristi za lokalne i orkestracione korake nad kojima CapabilityGraph provajdera nema nadležnost; *deferred* označava da iz postojećih metapodataka nije moguće bezbedno doneti odluku, na primer ako značenje identifikatora ili dinamičkog proizvođača još nije dovoljno formalizovano.

```
CapabilityValidationResult(  
  operation="get_lightcurve",  
  status="supported",  
  candidates=(  
    EndpointCapability(  
      broker="fink",  
      origin="lsst",  
      endpoint="sources"  
    ),  
  ),  
)
```

Provera se zasniva na strukturiranim dokazima, a ne na samim nazivima. Fizički poziv *sources* može zadovoljiti zahtev za svetlosnom krivom ako njegove *operation_types* i mapirani izlaz to potvrđuju. Fizički poziv nazvan *objects* može imati istu sposobnost, dok krajnja tačka čije ime sadrži *lightcurve* ne mora biti pogodna za traženu kardinalnost ili ograničenja. Slično tome, činjenica da odgovor izgleda kao niz nije dokaz da jedan poziv podržava više ciljnih identifikatora; za to mora postojati deklarirano koleksijsko povezivanje. Kompleksnost odnosa imena brokerskih metoda, sadržaja i struktura podataka takođe je ilustrovana u Tabeli 7.

9.7 Planiranje fizičke realizacije

Komponenta Planner vrši transformaciju validiranog WorkflowIR objekta u izvršni WorkflowRun. Tokom ovog procesa, za svaku poziciju unutar koraka (Step) generiše se odgovarajući StepRun, dok se operacije usmerene prema eksternim krajnjim tačkama izvršenja prevode u jedan ili više EndpointPlan objekata. Ovakav pristup odvajanju apstraktnog generičkog toka od njegove infrastrukturne konkretizacije predstavlja standardni model u savremenim sistemima za upravljanje naučnim radnim tokovima (Deelman et al., 2015).

```
WorkflowRun(
  workflow=workflow_ir,
  steps=(
    StepRun(
      step_index=0,
      state="planned",
      endpoint_plans=(EndpointPlan(...)),
    ),
    StepRun(
      step_index=1,
      state="planned",
      endpoint_plans=()
    )
  )
)
```

Jedan Step može imati nula fizičkih planova kada se izvršava lokalno, jedan plan kada ga zadovoljava jedan API endpoint ili više planova kada je potrebno više izvora ili fizičkih komponenti. Kod GetLightcurveStep-a za Fink/LSST optimizator može proizvesti obavezni plan za sources i neobavezni dopunski plan za fp. Oznaka required pripada odluci optimizatora: izvršni sloj ne pokušava da zaključuje da li je poziv obavezan na osnovu naziva API tačke izvršenja ili ponašanja servisa.

Step 2: GetLightcurveStep

```
EndpointPlan[0]: fink/lsst/sources required=True
EndpointPlan[1]: fink/lsst/fp required=False
```

Kada za jedno izvorno ograničenje postoji više ravnopravnih endpoint kandidata, a nije definisana politika njihovog rangiranja, Optimizator - Planner prijavljuje PlanningAmbiguityError umesto

da uvede skrivenu heuristiku. Kada kompatibilna sposobnost ne postoji, prijavljuje `UnsupportedStepError`; kada je odluka bezbedno odložena, `PlanningDeferredError`. Time plan ostaje objašnjiv i reproducibilan.

9.8 Pretvaranje kanonskih u fizičke parametre

Nakon planiranja, vrši se povezivanje kanonskih parametara Step-a sa fizičkim parametrima iz `EndpointSpec` ugovora. Deklaracija parametra navodi kanonsku ulogu, na primer `target_id`, `ra`, `dec` ili `radius`. Ime fizičkog parametra ostaje u registru provajdera i nikada ne ulazi u `WorkflowIR`.

```
# kanonska namera
GetLightcurveStep(
    target=TargetSelector(ids=["A", "B"], kind="object"),
    sources=[Source(broker="fink", origin="lsst")]
)

# fizički ugovor API-ja / krajnjih tačaka izvršenja
diaObjectId:
    bind: target_id
    binding:
        collection: csv
    type: string
    required: true

# rezultat povezivanja
params = {"diaObjectId": "A,B"}
```

Primer pokazuje da ni naziv `diaObjectId` ni zapis `A,B` nisu deo semantičkog Step-a.

`TargetSelector` čuva dva logička identifikatora, a generička kolekcijaska transformacija ih tek na granici poziva pretvara u reprezentaciju koju zahteva Fink/LSST. Drugi endpoint isti `target_id` može vezati na `objectId`, `objectIds`, `oid` ili numeričku vrednost, bez promene orkestracione namere.

9.9 Fizičko izvršenje i održavanje stanja

Izvršni sloj prima `BoundEndpointCall` i poziva registrovani transport. `EndpointRegistry` razrešava `EndpointSpec`, a `RegistryEndpointExecutor` proverava fizičke parametre, dopunjava

deklarisane podrazumevane i fiksne vrednosti, razrešava potrebne kredencijale, bira REST ili pozive preko Python klijenta i po potrebi primenjuje deklarisanu paginaciju. REST zahtevi koriste JSON, dok Python-client ugovor sadrži modul, naziv klijenta i metod umesto URL-a.

Rezultat jednog fizičkog poziva jeste `ExecutionResult`, sastavljen od sirovog odgovora i `InternalExecutionProvenance` zapisa. Tehnička provenijencija sadrži interni identifikator izvršenja, broker, poreklo, API endpoint, efektivne parametre, vreme početka i završetka, transport, metod, sanitizovana zaglavlja i dostupne metapodatke odgovora. Ona ne zamenjuje semantičku provenijenciju pojedinačnih normalizovanih podataka (Moreau & Missier, 2013; Servillat et al., 2020).

```
ExecutionResult(
    payload=raw_provider_response,
    execution_provenance=InternalExecutionProvenance(
        internal_execution_id="exec:...",
        broker="fink",
        origin="lsst",
        endpoint="sources",
        params={"diaObjectId": "A,B"},
        status="success",
        transport="rest",
        response_status_code=200
    )
)
```

StepRun vodi minimalno stanje jedne pojave Step-a: *pending*, *planned*, *running*, *succeeded* ili *failed*. Pored toga čuva `EndpointPlan` objekte, identifikatore uspešnih izvršenja, poravnanje izvršenja sa indeksima planova, eventualno prazne planove zavisne od kandidata, upozorenja i opis greške. Kod obaveznih planova izvršenje se prekida pri prvoj grešci (engl. *fail-fast*; Gray, 1986). Neobavezni dopunski plan može prijaviti upozorenje i dozvoliti uspeh Step-a ako je njegova osnovna semantička potreba već zadovoljena.

9.10 Normalizacija i kardinalnost rezultata

Po uspešnom fizičkom izvršenju sirovi odgovor se normalizuje deklarativnim mapiranjima, definisanim u datotekama *mappings.yaml* posebno za svaki par proizvođač-broker. Važno je da

kardinalnost izvršenja i kardinalnost semantičkih objekata nisu iste. Jedan poziv sa dva ciljna identifikatora predstavlja jedno fizičko izvršenje, ali može proizvesti dva Portfolio objekta. Jedan Portfolio i dalje reprezentuje tačno jedan primarni astronomski objekat.

Semantički **ulaz**:

```
target.ids = ["A", "B"]
```

Fizičko **povezivanje**:

```
diaObjectId = "A,B"
```

Fizičko izvršenje:

```
1 EndpointPlan
1 BoundEndpointCall
1 ExecutionResult
```

Normalizovani rezultat:

```
ExecutionPortfolioResult(
  portfolios=(Portfolio_A, Portfolio_B)
)
```

Na primer, zahtev ka Fink brokeru za LSST podatke može sadržati parametar diaObjectId, dok se unutar odgovora isti identitet mapira pod ključem r:diaObjectId. Kod nekog drugog provajdera, poziv i odgovor mogu koristiti potpuno drugačije strukturne putanje i konvencije imenovanja. Ova vrsta ulazne heterogenosti i sintaksne konfuzije efikasno se rešava na nivou medijatorskog sloja putem deklarativnih ugovora unutar *endpoints.yaml* datoteke. To je pristup izolaciji eksternih varijacija koji se oslanja na klasičnu arhitekturu inteligentnih medijatora (Wiederhold, 1992), kao i na principe upravljanja zasnovanog na modelima (engl. *model-based data management*), gde se prevođenje različitih šema vrši bez uticaja na unutrašnji semantički model, koji ostaje stabilan (Bernstein & Melnik, 2007).

ExecutionPortfolioResult čuva nula ili više Portfolio objekata nastalih iz jednog fizičkog izvršenja. StepPortfolioResult čuva sva fizička izvršenja pripadajuća jednoj pojavi Step-a i iz njih izvodi semantički pogled tog koraka. WorkflowPortfolioResult povezuje završni WorkflowRun sa uređenim nizom svih StepPortfolioResult objekata. Ovako se istovremeno čuvaju fizička revizionna struktura i semantički pogled toka.

9.11 Akumulacija semantičkog materijala

Kada kasniji Get korak obogaćuje ranije kandidate, novi fizički odgovori pripadaju tom kasnijem Step-u. Istovremeno, njegov semantički pogled može sadržati raniji Portfolio materijal i novodobijene zapise. Za to StepPortfolioResult koristi *materialized_portfolios*: nepromenljivi snimak akumuliranog materijala u toj tački toka.

Step 0: Search

executions: search execution
portfolios: summary + detection

Step 1: GetClassification

executions: classification execution
portfolios: summary + detection + classification

Step 2: GetLightcurve

executions: lightcurve execution
portfolios: summary + detection + classification + lightcurve

Istorijski pogled Step-a 0 se ne menja kada se izvrši Step 1 ili Step 2. Ovaj princip nepromenljivosti prethodnih rezultata je važan da bi bilo moguće rekonstruisati šta je bilo dostupno posle svake operacije, a da se ranija izvršenja ne pripisuju kasnijim Step-ovima. Portfolio objekti se konsoliduju samo kada se nedvosmisleno utvrdi isti par (*origin*, *object_id*). Neidentifikovani i dvosmisleni rezultati ostaju odvojeni. Ovo harmonizovanje je automatska operacija i ne treba ga brkati sa Match korakom koji je naučna, semantička operacija.

9.12 Naknadno povezivanje koraka i identiteta kandidata

Ne mogu svi fizički pozivi uvek biti povezani unapred. Kasniji Step često zahteva identifikatore objekata koji postaju poznati tek kada se raniji poziv izvrši, a odgovor normalizuje i eventualno lokalno filtrira ili poveže; *execute_staged_workflow_run* zato prolazi kroz WorkflowRun po redosledu Step-ova i povezuje svaki poziv kada su njegove vrednosti dostupne u vremenu izvršenja (Deelman et al., 2015).

Step 0 SemanticSearchStep

-> izvršenje i normalizacija

-> kandidati A, B, C

Step 1 FilterStep

-> lokalni predikat

-> kandidati A, C

Step 2 GetLightcurveStep

-> late binding `target_id = [A, C]`

-> izvršenje i normalizacija

-> obogaćeni **Portfolio** A i C

9.13 Ponovna upotreba fizičkog izvršenja

Dva različita semantička Step-a ponekad mogu biti zadovoljena istim fizičkim odgovorom. Na primer, pretraga sa uslovom nad klasifikacijom može pozivati krajnju tačku izvršenja koja u odgovoru već obavezno proizvodi traženi *classification* zapis. Ako sledeći `GetClassificationStep` traži upravo taj nekvalifikovani ili odgovarajuće kvalifikovani zapis, `Planner` može postaviti `execution_reuse_from` umesto novog poziva.

Odluka je dozvoljena samo kada se poklapaju fizički API poziv, potrebna semantička porodica i relevantni kvalifikatori, a predikat ili mapiranje pružaju dovoljan dokaz. Pri izvršenju se zatim koristi keširani `ExecutionResult` ranijeg plana. Normalizacija čuva isti `execution ID` i osnovne `Portfolio` identitete, ali za svaki Step može primeniti njegov rezidualni predikat i izgraditi poseban semantički pogled. Tako optimizacija fizičkog rada ne briše semantičku strukturu toka.

9.14 Inkrementalno proširivanje aktivnog toka

Orkestraciono jezgro obezbeđuje integritet nastavka delimično izvršenog toka, pritom ostajući potpuno nezavisno od specifičnosti korisničkog interfejsa. Bilo da se radi o novom DSL fragmentu, vizuelnoj operaciji ili pozivu kroz programski klijent, svaki novi unos generiše prošireni `WorkflowIR`. Međutim, da bi se nastavak izvršavanja proglasio validnim i bezbednim, moraju biti ispunjena dva stroga uslova:

- Strukturni kontinuitet: Prethodni `WorkflowIR` mora ostati tačan, nepromenjen prefiks novog, proširenog toka;

- Invarijantnost istorije: Planovi izvršenja prethodnih koraka (Step objekata) ne smeju biti retroaktivno modifikovani.

Ukoliko bilo koji od ovih uslova nije zadovoljen, orkestraciono jezgro odbija inkrementalno nastavljanje i zahteva započinjanje potpuno novog izvršnog procesa. Takav rigidni model obezbeđivanja integriteta toka implementira principe upravljanja naučnom provenijencijom i keširanja međurezultata koje su definisali Freire et al., 2006.

`previous.run.workflow`

`-> WorkflowIR(steps=[Search, Filter])`

new fragment

`-> append GetLightcurve`

extended `WorkflowIR`

`-> WorkflowIR(steps=[Search, Filter, GetLightcurve])`

Tokom inkrementalnog izvršenja stari fizički pozivi se ne šalju ponovo provajderima. Njihovi sačuvani rezultati ponovo se koriste na istom mestu u obradi na kojoj bi se pojavio novi `ExecutionResult`, uz proveru identiteta tačaka izvršenja i povezanih parametara. Tek nakon potpunog poklapanja starog prefiksa novi pozivi se delegiraju izvršenju. Planiranje i normalizacija se ponavljaju nad celim kumulativnim tokom kako bi rezultat ostao potpun i nepromenljiv semantički snimak. U implementaciji, povezanoj sa korisničkim interfejsom, ovo načelo se primenjuje tako što se aktivno stanje jedne početne UI sesije čuva kao `StagedWorkflowResult`.

9.15 Rezultat, prikaz i provenijencija

Završni rezultat toka nije sirovi odgovor poslednjeg poziva API metode. On obuhvata `WorkflowRun`, Step rezultate i završni `Portfolio` pogled nakon poslednje operacije. Javni API može taj rezultat serijalizovati u JSON ili ga proslediti HTML prikazu, grafičkom interfejsu ili izvoznom mehanizmu. Instrukcije prikaza, kao što je `ResultViewSpec` za sortiranje rezultata, nalaze se van `WorkflowIR`-a zato što ne menjaju naučni sadržaj `Portfolio` objekata.

Reproducibilnost se ostvaruje spajanjem nekoliko eksplicitnih zapisa. `WorkflowIR` čuva semantički redosled operacija. `WorkflowRun` čuva izabrane fizičke planove, njihove zavisnosti i stanja. `InternalExecutionProvenance` čuva konkretne pozive, parametre, transport i odgovor.

SemanticRecord i SemanticEdge čuvaju domenske podatke - normalizovani semantički materijal. StepPortfolioResult čuva kako se semantički pogled menjao kroz tok rada. Zahvaljujući ovoj podeli, svaka od navedenih struktura ima zasebnu ulogu u praćenju izvršenja. Ovaj distribuirani pristup oslikava zahteve za semantičkom transparentnošću (Gil et al., 2007) i usklađen je sa savremenim standardima i formalnim modelima za reprezentaciju provenijencije podataka (Moreau & Missier, 2013).

Ovakva organizacija omogućava da se odgovori na različita pitanja: šta je naučni tok zahtevao, zašto je izabran određeni fizički poziv, koji parametri su prosleđeni, koji je odgovor dobijen, kako je normalizovan i koji su kandidati preživeli svaku lokalnu operaciju. Istovremeno, model ostaje uži od potpunog sistema distribuiranog praćenja, (engl. *distributed tracing*, Sigelman et al., 2010); cilj Alertissima je proverljivo naučno i tehničko poreklo, dok je kod sistema distribuiranog praćenja cilj beleženje svakog internog događaja procesa.

9.16 Trenutni obuhvat i ograničenja izvršnog sloja

Aktuelna implementacija realizuje celovit put za vremenski ograničene request/response tokove: formalni WorkflowIR model koraka, validaciju mogućnosti, determinističko planiranje, deklarativno povezivanje, REST i Python-client izvršenje, normalizaciju podataka, Portfolio konsolidaciju, rezidualno filtriranje, reproducibilni radni tok, akumulaciju materijala i tehničku provenijenciju. Implementirani su i lokalni Filter, pozicioni Match, brokerski Confirm kvorum, kao i pojedine lokalne derivacije. Podržano je da jedan Step ima više fizičkih planova, da jedan poziv proizvede više Portfolio objekata i da kasniji Step koristi kandidate ili materijal ranijih operacija.

IR rečnik je namerno širi od trenutnog podskupa operacija za koje postoji izvršna podrška.

GetSpectrum nema registrovanu fizičku realizaciju u sadašnjem brokerskom skupu. Opšte Analyze i Action operacije još nemaju objedinjene izvršiće. Lokalna konstrukcija pomoću LightcurveStep-a nije završena, iako GetLightcurveStep i normalizacija svetlosnih krivih jesu funkcionalni. Nije implementirana podrška za praćenje toka u realnom vremenu putem MonitorStep-a.

Ova ograničenja pokazuju gde se završava dokazana implementacija, ali ne menjaju osnovnu podelu odgovornosti. Najvažniji rezultat orkestracionog sloja je činjenica da se semantička namera, fizički plan, povezani poziv, izvršenje i normalizovani materijal mogu razvijati i proveravati kao odvojene, formalno povezane reprezentacije.

10 Sloj domenskog jezika - DSL

Alertissimo DSL predstavlja korisnički sloj za deklarativno opisivanje naučnih tokova nad heterogenim brokerskim servisima. Njegov zadatak nije da pred korisnika izlaže pojedinačne API pozive, već da omogući iskazivanje namere: koji skup objekata treba formirati, kojim podacima ga treba obogatiti, koje uslove treba proveriti i koje operacije treba izvršiti nad dobijenim semantičkim materijalom. Fizička realizacija te namere - izbor mogućnosti provajdera, povezivanje parametara, poziv REST servisa ili Python klijenta i normalizacija odgovora - pripada slojevima opisanim u poglavlju 9.

Ovakva podela neposredno sledi osnovni cilj sistema Alertissimo: isti naučni scenario treba da ostane razumljiv i stabilan i kada se promeni servis preko kojeg se podaci pribavljaju. U prvom objavljenom prikazu sistema (Vujčić et al., 2026) orkestracija brokerskih tokova postavljena je kao način da se razlike između provajdera prevaziđu na nivou sistema. Aktuelni DSL tu ideju formalizuje kao izvršiv jezik: korisnički iskaz se parsira, semantički i operaciono validira, prevodi u kanonski WorkflowIR, a tek zatim planira i izvršava.

10.1 Uloga i projektni ciljevi DSL sloja

Mernik, Heering i Sloane (2005) definišu domenski specifičan jezik kao jezik prilagođen određenoj oblasti primene, sa većom izražajnošću unutar te oblasti nego što je pruža opšti programski jezik. Ta ideja je za Alertissimo važna iz praktičnog razloga: istraživač treba da govori o objektima, prostornim i vremenskim ograničenjima, svetlosnim krivama, klasifikacijama, povezivanju i potvrđivanju kandidata, a ne o URL adresama, HTTP metodama, programabilnim interfejsima, imenima parametara i strukturi brokerskog JSON odgovora.

Van Deursen, Klint i Visser (2000) pokazuju da je vrednost DSL-a najizraženija kada njegov rečnik odgovara pojmovima problema, dok se implementacioni detalji zadržavaju izvan korisničkog izraza. U Alertissimu se to načelo ostvaruje kroz četiri projektna cilja.

Prvo, jezik je deklarativan. Skript opisuje željeni naučni tok, ali ne propisuje redosled mrežnih poziva, način paginacije, konstrukciju ili format rezultata. Drugo, jezik je nezavisan od fizičkog ugovora servisa: naziv proizvoda u DSL-u ne mora da odgovara nazivu konačne tačke brokerskog izvršenja niti fizičkom nazivu parametra. Treće, isti formalni oblik jezika služi kao ulaz u validaciju

i prevođenje u IR, pa korisnički iskaz nije slobodan tekst koji izvršni sloj naknadno pokušava da protumači. Četvrto, DSL je stabilna meta i za buduće interfejse: prirodni jezik ili vizuelni alati treba da proizvedu isti formalni DSL, umesto da neposredno upravljaju izvršenjem.

Najjednostavniji potpuni program formira skup kandidata iz jednog ili više pregleda neba preko izabranog brokera:

```
objects from ztf via lasair  
inside (305.5822, -18.7909, 300arcsec)
```

Ovaj kratki program ne znači „pozovi unapred određeni Lasair endpoint“. On znači: formiraj početni skup objekata poreklom iz pregleda ZTF u zadatoj oblasti neba, uz ograničenje da se za ovu nameru koristi mogućnost koju objavljuje Lasair. Validacija softverske mogućnosti proverava da li takva realizacija postoji, a optimizator tek posle prevođenja u IR bira fizičku operaciju.

10.2 Od čitljive sintakse do proverljivog toka

Sintaksa namerno koristi kratke izraze nalik upitnom jeziku. Chamberlin i Boyce (1974) su još u jeziku SEQUEL pokazali vrednost strukturisanih, engleskom jeziku bliskih konstrukcija za deklarativno opisivanje zahteva nad podacima. Alertissimo preuzima čitljivost takvog pristupa, ali ne preuzima relacionu semantiku SQL-a. SQL i njegovi astronomski derivati opisuju upit nad tabelama unutar poznatog sistema, dok Alertissimo DSL opisuje naučni scenario sastavljen iz više kompleksnih koraka čija fizička realizacija može obuhvatiti različite servise i lokalne operacije.

Razlika se jasno vidi u poređenju sa ADQL-om. Demleitner et al. (2023) definišu ADQL 2.1 kao SQL92-zasnovan jezik sa astronomskim geometrijskim proširenjima za upite nad servisima Virtuelne opservatorije (IVOA). Takav jezik je prirodan kada je cilj projekcija, selekcija i spajanje tabelarnih podataka. Alertissimo DSL, nasuprot tome, može da započne pronalaženjem kandidata kod jednog brokera, nastavi pribavljanjem proizvoda kod drugog, lokalno suzi skup i zatim za preostale objekte pokrene naredni brokerski korak. Zato sličnost sa SQL-om ostaje na nivou deklarativnosti i čitljivosti, ne i na nivou modela izvršenja.

Slična granica postoji i prema semantičkim jezicima upita. Harris i Seaborne (2013) u specifikaciji SPARQL 1.1 definišu sintaksu i semantiku upita nad RDF grafovima. Alertissimo koristi ontološke pojmove i semantičke putanje, ali njegov DSL nije SPARQL dijalekt: predmet jezika nije

proizvoljno pretraživanje RDF grafa, već konstrukcija validnog astronomskog toka rada koji može kombinovati udaljeno i lokalno izvršenje.

Potpuniji primer ilustruje osnovnu nameru jezika:

```
objects from ztf via lasair
inside (305.5822327501884, -18.7909207179724, 300arcsec)
latest 20
with lightcurve via fink
```

Program formira početni skup ZTF objekata (ograničen na 20) u zadatoj oblasti neba preko sistema Lasair, zatim zahteva svetlosnu krivu preko sistema Fink. U njemu nema fizičkog naziva parametra za koordinatu, načina kodiranja radijusa, imena Fink metode niti strukture odgovora. Upravo je odsustvo tih detalja uslov da ista namera može biti ponovo planirana kada se promene deklaracije softverskih mogućnosti ili fizički ugovori servisa.

10.3 Mesto DSL-a među semantičkim reprezentacijama

DSL nije jedina semantička reprezentacija u Alertissimu. On predstavlja korisnički jezički sloj, dok WorkflowIR predstavlja kanonsku specifikaciju toka za orkestraciju, a semantički model sa ontologijom određuje značenje podataka preko hijerarhijskih putanja i nivoa te definiše sistemske strukture podataka. Ove tri reprezentacije se preklapaju, ali nisu međusobno zamenljive.

DSL čuva jezičke izbore prilagođene čoveku, uključujući oblike jednine i množine, redosled klauzula i specifične ključne reči. Površinski AST (engl. *abstract syntax tree*) transformiše ovaj unos u formalnu sintaktičku strukturu, bez razdvajanja naučnog i čisto reprezentacionog izraza korisnika (poput klauzule `order by`). WorkflowIR vrši apstrakciju ovih sintaktičkih elemenata i prevodi ih u čiste semantičke namere predstavljene kroz Step objekte, praveći pomenutu razliku, kao i semantičko razdvajanje lokalnih derivaciono-analitičkih i distribuiranih koraka; ovakva forma je i dalje potpuno agnostična prema konkretnim pozivima izvršenja, što optimizatoru izvršenja omogućava da radni tok samostalno interpretira prema brokerskim ugovorima. Konačno, ontologija obezbeđuje semantički kontekst koji sirove povratne podatke i mrežne putanje razrešava u stabilni, standardizovani prostor domenskih atributa. Celovit pregled generalne arhitekture dat je na Slici 7.

10.4 Sintaksa, gramatika i procesiranje DSL-a

Obrada i interpretacija DSL koda u sistemu Alertissimo definisani su kroz spregu formalne gramatike, leksičkih invarijanti i procesa parsiranja. Parser i validatori prevode tekstualni opis toka u strogo tipizirane modele, prateći pravila i mehanizme izložene u nastavku.

10.4.1 Formalna gramatika i parsiranje

Jezik je definisan formalno napisanom Lark gramatikom. Glavna gramatika opisuje strukturu programa i klauzula, dok odvojena gramatika izraza opisuje logičke predikate. Ovo razdvajanje je namerno: proširenje jezika novom klauzulom ne treba automatski da menja pravila za logičke operatore, a proširenje izraza novim operandima ne treba da naruši strukturu programa.

Sintaktički analizator (parser) operiše u LALR režimu i koristi integrisani kontekstualni leksički analizator (engl. *contextual lexer*). Prema zvaničnoj dokumentaciji alata Lark, primenjeni LALR²² algoritam garantuje linearnu vremensku složenost ($O(n)$) i minimalnu memorijsku potrošnju tokom obrade nedvosmislenih gramatika. U kontekstu interaktivnog DSL-a, sistem se oslanja na tri povezane komponente alata Lark:

- Kontekstualno skeniranje: Lekser prima povratnu informaciju od parsera i dinamički sužava podskup dozvoljenih terminala na one koji su legalni u datom stanju, čime se eliminišu kolizije imena²³;
- Automatsko praćenje pozicije: Svaki generisani token automatski zadržava metapodatke o tačnom redu i koloni izvornog teksta;
- Interaktivno upravljanje greškama: U slučaju sintaktički neispravnog unosa, mehanizam trenutno prekida proces i izbacuje izuzetak sa detaljima greške.

Ovakav dizajn omogućava da se formalna greška detektuje i korisniku pruži povratna informacija direktno na mestu nastanka u izvornom skriptu, umesto da se manifestuje kao neodređena ili kaskadna greška tokom kasnijih faza izvršenja radnog toka.

²² <https://lark-parser.readthedocs.io/en/latest/features.html>

²³ <https://lark-parser.readthedocs.io/en/latest/parsers.html>

10.4.2 Pravila rasporeda i oblikovanja klauzula

Potpuni program mora započeti iskazom koji uvodi kandidate. Posle njega slede klauzule koje su razdvojene novim redom. Uvlačenje služi samo preglednosti i nema semantiku bloka:

```
objects from lsst via alerce
inside (305.5822, -18.7909, 30arcsec)
latest 5
with lightcurve
```

Isti program može biti napisan bez uvlačenja i proizvodi istu površinsku strukturu. Takvo pravilo sprečava da razmak nevidljivo promeni naučno značenje. Grupisanje logičkih uslova uvek se izražava operatorima i zagradama.

Komentar počinje znakom # i traje do kraja reda:

```
# Pronalaženje kandidata u sfernoj oblasti neba
objects from ztf via alerce
inside (305.5822, -18.7909, 60arcsec) # prostorno ograničenje
with lightcurve via fink
```

Pre parsiranja se raspored redova kanonizuje tako da se ukloni semantički nebitno uvlačenje, ali se sačuva broj redova. Na taj način prijavljena lokacija greške ostaje korisna u editoru. Parser zatim proizvodi strogo tipiziran i nepromenljiv površinski model.

Pored same gramatike postoje i pravila rasporeda klauzula. Dozvoljena je samo jedna opšta where klauzula; ona mora prethoditi with i filter klauzulama. Prostorna, vremenska i latest ograničenja pripadaju početnom formiranju kandidata. Kasnije where klauzule moraju pripadati konkretnim with i filter klauzulama. Ova pravila nisu estetska: ona sprečavaju da isti tekst dobije različito značenje u zavisnosti od implicitnog stanja tokom izvršenja.

10.5 Formiranje početnog skupa kandidata

Početni DSL iskaz ima dve osnovne forme: otkrivanje kandidata na osnovu predikata i razrešavanje poznatih identifikatora. Njihovo razdvajanje zadržava razliku između pitanja „koji objekti ispunjavaju uslov“ i pitanja „pribavi podatke za ove već poznate objekte“.

10.5.1 Otkrivanje kandidata

Pretraga počinje množinom `objects`, listom jednog ili više opservacionih porekla i opcionim izborom brokera:

```
objects from lsst, ztf via alerce
```

Poreklo i broker nisu sinonimi. `lsst` i `ztf` označavaju opservaciono poreklo podataka, dok `alerce` označava servis preko kojeg se zahteva realizacija. Ovim se i na jezičkom nivou čuva razlika između proizvođača naučnog materijala i provajdera koji taj materijal dostavlja.

Ako `via` nije naveden, izbor provajdera ostaje otvoren planeru, ali skript i dalje mora proći validaciju softverskih mogućnosti nad raspoloživim grafom. DSL zato ne obećava da će svaka apstraktna kombinacija biti izvršiva; on omogućava da se takva kombinacija formalno iskaže i proveriti pre mrežnog poziva.

10.5.2 Razrešavanje poznatih identifikatora

Aktuelna implementacija podržava iskaze pretrage po identifikatoru za objekte i alert događaje. Jednina označava jedan identifikator:

```
object ZTF20aafqubg from ztf via antares  
with lightcurve via antares
```

Množina prihvata listu jedinstvenih identifikatora:

```
objects ZTF20aafqubg, ZTF18acurdih from ztf via antares  
with lightcurve via lasair
```

Analogno, `alert` i `alerts` razrešavaju identifikatore pojedinačnih alert događaja. Površinski model proverava da oblik jednine sadrži tačno jedan identifikator, a da lista nije prazna i da nema duplikata. Zatim se konstruiše `LookupStep` sa `TargetSelector` objektom vrste `object` ili `alert`.

`Lookup` nije pretraga sa praznim predikatom. Pošto su kandidati već određeni identifikatorima, početne klauzule `inside`, `within`, `latest` i opšti `where` nisu dozvoljene uz `lookup` iskaz.

10.6 Prostorna, vremenska i količinska ograničenja

Klauzula `inside` izražava kružnu oblast preko rektascenzije, deklinacije i ugaonog radijusa:

```
objects from lsst via alerce
inside (305.5822327501884, -18.7909207179724, 1.5arcmin)
```

Formalni model proverava da je rektascenzija veća ili jednaka 0° , a manja od 360° , da je deklinacija između -90 i 90 stepeni i da je radijus pozitivan. Ove provere se odnose na kanonski smisao koordinata. Pretvaranje u fizičke nazive parametara i jedinice konkretnog servisa ostaje odgovornost bindera.

Klauzula `within` ima relativnu i apsolutnu formu. Relativna forma izražava trajanje u sekundama, minutima, satima, danima ili nedeljama:

```
objects from ztf via lasair
inside (305.5822, -18.7909, 300arcsec)
within 7d
```

Apsolutna forma navodi početnu i krajnju granicu:

```
objects from ztf via lasair
inside (305.5822, -18.7909, 300arcsec)
within 2026-01-01T00:00:00Z, 2026-01-31T23:59:59Z
```

Ovo razdvajanje odgovara razlici između intervala vezanog za trenutak izvršenja i unapred poznatog vremenskog okvira. Allen (1983) je pokazao da je eksplicitno modelovanje vremenskih intervala važno za pouzdano zaključivanje o vremenskim odnosima; Alertissimo DSL u sadašnjem obimu ne uvodi čitavu algebru intervala, ali čuva granice okvira kao semantički vremenski kontekst.

Klauzula `latest` ograničava početni skup na pozitivan broj najnovijih kandidata:

```
objects from ztf via alerce
inside (305.5822, -18.7909, 300arcsec)
latest 10
```

`latest` pripada početnom prolazu formiranja kandidata. Nije zamena za `order by`: prva klauzula utiče na izbor kandidata, dok druga određuje način prikaza već proizvedenog rezultata.

10.7 Semantički predikati

Predikati se pišu u posebnom izražajnom podjeziku. Podržani su operatori poređenja `=`, `!=`, `<`, `<=`, `>` i

$\geq$, Bulove vrednosti, tekstualni literali, brojevi, operatori `not`, `and` i `or`, zagrade i provera postojanja `exists`. Prioritet je eksplicitan: `not` se primenjuje pre `and`, a `and` pre `or`.

Jednostavan predikat nad normalizovanim kvalitetom detekcije izgleda ovako:

```
filter detection@ztf:fink.quality.real_bogus >= 0.8
```

Referenca se sastoji od vrste semantičkog zapisa, opcionog kvalifikatora i putanje polja. U izrazu `detection@ztf:fink.quality.real_bogus`, `detection` je vrsta zapisa, `ztf` je proizvođač odnosno poreklo tvrdnje, `fink` je kanal preko kojeg je zapis pribavljen, a `quality.real_bogus` je relativna semantička putanja unutar zapisa. Kvalifikator je važan kada Portfolio sadrži više zapisa iste porodice iz različitih izvora.

Provera postojanja omogućava da se odsustvo podatka razlikuje od konkretne vrednosti:

```
filter exists classification@lc_classifier:alerce.best.class
```

Složeniji uslov koristi eksplicitno grupisanje:

```
filter (classification@lc_classifier:alerce.best.class = "SN" and  
classification@lc_classifier:alerce.best.probability >= 0.7) or  
classification@lc_classifier:alerce.best.class = "AGN"
```

Parser ne koristi uvlačenje da bi odredio opseg operatora. Zagrade i Bulovi operatori jedini su izvor strukture, pa isti uslov ima stabilno značenje u editoru, API pozivu i sačuvanom skriptu.

Semantička validacija zatim proverava da referencirani tip zapisa postoji u ontologiji, da putanja može biti razrešena kroz hijerarhijski model i da je terminalna vrednost pogodna za datu operaciju.

10.8 Razlika između *where* i *filter*

I `where` i `filter` sadrže semantički predikat, ali pripadaju različitim fazama toka: `where` opisuje uslov početnog pronalaženja kandidata (optimizator kasnije može utvrditi da li se ceo predikat može potisnuti ka provajderu, da li deo mora ostati rezidualni lokalni uslov ili da li zahtev nije podržan); `filter` je eksplicitna lokalna operacija nad već akumuliranim, normalizovanim semantičkim materijalom.

Opšti uslov pretrage piše se pre obogaćivanja:

```
objects from lsst via alerce
```

```
inside (305.5822, -18.7909, 300arcsec)
where classification@stamp_classifier_rubin_beta_20260421.best.class = "SN" and
classification@stamp_classifier_rubin_beta_20260421.best.probability >= 0.5
```

Ovde where učestvuje u definisanju početnog skupa kandidata. Semantička referenca istovremeno uvodi potrebu za odgovarajućim klasifikacionim materijalom, a validacija mogućnosti proverava postoji li dozvoljena realizacija za konkretan izvor.

Nasuprot tome, sledeći primer najpre pribavlja svetlosnu krivu preko jednog brokera, zatim lokalno zadržava kandidate koji zadovoljavaju uslov i tek za preostale objekte pokreće naredno obogaćivanje:

```
objects from ztf via lasair
inside (305.5822, -18.7909, 300arcsec)
with lightcurve via fink
filter detection@ztf:fink.quality.real_bogus >= 0.8
with lightcurve via lasair
```

Razlika nije samo sintaktička: filter postaje lokalni FilterStep, pa njegov rezultat menja skup ciljnih identifikatora za naredni korak. Tako DSL neposredno izražava fazno izvršavanje: udaljeno pribavljanje, lokalna semantička odluka i nastavak rada samo nad preostalim kandidatima. Where – filter dualitet eksplicitno je podržan zbog koncepta nastavka, tj. inkrementalnog toka rada nakon početnog korisničkog pregleda rezultata.

10.9 Pravci proširenja gramatike selekcije

Jedno korisno proširenje, koje još nije deo aktuelne gramatike ni izvršnog modela, jeste klauzula having za filtriranje nakon materijalizacije podataka. Broj tačaka po fotometrijskom filtru, na primer, poznat je tek nakon pribavljanja i normalizacije svetlosne krive:

```
objects from lsst via alerce
inside (305.5822, -18.7909, 300arcsec)
with lightcurve via alerce
having count(lightcurve.points) >= X
and every band in ["u", "g", "r", "i", "z", "y"] satisfies
  count(lightcurve.points where band = $band) >= 5
```

U ovom predlogu where zadržava postojeću ulogu: filtrira pretraživa svojstva objekta pre ili tokom pronalaženja kandidata. Klauzula having vrednovala bi se tek pošto su zahtevani proizvodi materijalizovani i omogućila bi uslove nad kolekcijama, agregatima i rezultatima obogaćivanja. Drugi uslov iz primera već zahteva najmanje 30 tačaka, po pet u svakom od šest LSST filtara, pa

`count(...)` $\geq X$ uvodi dodatno ograničenje samo kada je $X > 30$.

Pre uvođenja takve sintakse mora se precizirati šta se računa kao posmatranje. Ako gornje granice, forsirana fotometrija ili nedetekcije ne treba da uđu u zbir, kriterijum mora to izričito navesti:

```
having count(lightcurve.points where measurement.valid = true) >= X
and every band in ["u", "g", "r", "i", "z", "y"] satisfies
    count(lightcurve.points where band = $band
        and measurement.valid = true) >= 5
```

Semantički, ova konstrukcija mogla bi da se prevede u generički agregatni i kvantifikovani predikat nad normalizovanom kolekcijom `lightcurve.points`. Planer bi ga potisnuo ka servisu samo kada deklaracija softverskih mogućnosti potvrđuje potpuno isti agregat; u suprotnom bi najpre pribavio svetlosne krive i isti predikat primenio rezidualno. Za to nije potrebna posebna planerska grana za svakog provajdera. Nazivi `band` i `measurement.valid` u primeru imaju ilustrativnu ulogu i morali bi, zajedno sa sintaksom `having`, da budu formalizovani u budućoj verziji modela.

10.10 Zahtevi za semantičkim obogaćivanjem

Klauzula `with` zahteva dodatni semantički proizvod ili zapis. Osnovna forma navodi proizvod:

```
objects from ztf via alerce
inside (305.5822, -18.7909, 60arcsec)
with lightcurve
```

Izraz `lightcurve` ne razrešava se poređenjem sa nazivom tačke izvršenja. Semantička validacija ga razrešava u tačno jednu porodicu zapisa ontologije, validacija mogućnosti traži podržanu realizaciju, i konstruiše se odgovarajući `GetLightcurveStep`. Ako je izraz dvosmislen ili nepoznat, greška se prijavljuje pre planiranja.

Opcionim delovima razdvajaju se proizvođač, kanal izvršenja i metoda:

```
with classification from stamp_classifier_rubin_beta_20260421 via alerce
```

`from` sužava proizvođača semantičke tvrdnje, `via` bira provajdera, a `using` rezerviše mesto za imenovanu metodu kada proizvod dopušta lokalno izvođenje. Ove uloge ne treba objediniti u jedno polje: klasifikator koji proizvodi tvrdnju i broker koji tu tvrdnju isporučuje mogu biti različiti sistemi. U aktuelnoj implementaciji `using` se koristi pri lokalnoj klasifikaciji:

```
objects from ztf via alerce
inside (305.5822, -18.7909, 60arcsec)
```

```
with classification using alertissimo:clasMeV2
```

Nakon parsiranja, ovakav zahtev prevodi se u `ClassifyStep` i čuva izabranu metodu. Aktuelni planer i izvršni sloj još nemaju registrovanu realizaciju lokalnog klasifikatora, pa primer pokazuje podržanu sintaksu i IR reprezentaciju, ali ne i potpuno izvršiv tok. Kod udaljenog preuzimanja svetlosne krive, forsirane fotometrije, isečka slike ili spektra kvalifikator `using` nije podržan; prijavljuje se kontrolisana greška `unsupported_requirement_method`.

Predikat može biti vezan neposredno za zahtev. U tom slučaju relativne putanje se tumače u kontekstu proizvoda iz iste `with` klauzule:

```
objects from lsst via alerce
inside (305.5822, -18.7909, 300arcsec)
with classification from stamp_classifier_rubin_beta_20260421 via alerce where
best.class = "SN" and best.probability >= 0.5
```

Putanje `best.class` i `best.probability` ne moraju da ponavljaju puni naziv zapisa i kvalifikator jer ih daje okolna klauzula. Ovakva skraćena forma ostaje formalna: validator je širi u tačno određeni semantički kontekst pre prevođenja u međureprezentaciju toka rada.

Jedan tok može zahtevati isti proizvod preko više provajdera:

```
objects from ztf via alerce
inside (305.5822, -18.7909, 300arcsec)
with lightcurve via fink
with lightcurve via lasair
```

To nije redundantno. Dve klauzule predstavljaju dve semantičke pojave obogaćivanja i mogu doprineti različitim detekcijama, nedetekcijama, forsiranoj fotometriji ili provenijenciji. Normalizacija ih dovodi u zajednički model, ali ne briše poreklo podataka.

10.11 Potvrđivanje, povezivanje i prikaz rezultata

Klauzula `confirm` zahteva potvrdu kandidata kvorumom navedenih brokera:

```
objects from ztf via alerce
inside (305.5822, -18.7909, 300arcsec)
latest 1
confirm by 2 via fink, alerce, antares
with lightcurve via fink
```

Površinski model proverava da je zadati broj potrebnih potvrda pozitivan, da se brokeri ne ponavljaju i da kvorum nije veći od broja navedenih brokera. Ovom transformacijom nastaje `ConfirmStep`; fizičko pribavljanje dokaza i lokalna provera kvoruma pripadaju orkestracionom i

izvršnom sloju.

Klauzula `match` izražava naučno povezivanje kandidata. Poziciona varijanta koristi eksplicitni kriterijum:

```
objects from lsst, ztf via alerce
inside (305.5822, -18.7909, 300arcsec)
match on position inside 1arcsec
```

Ovde početni `inside` određuje oblast pretrage, dok `match on position inside 1arcsec` određuje odnos između parova kandidata.

`order by` klauzula ne proizvodi naučni `Step`. Ona opisuje prikaz rezultata i izdvaja ga u `ResultViewSpec`:

```
objects from ztf via lasair
inside (305.5822, -18.7909, 300arcsec)
with lightcurve via fink
order by summary.time.last_mjd desc
```

Ova odluka čuva reproducibilnost toka rada: promena redosleda prikaza ne menja korake koji su proizveli semantički materijal niti tehničku provenijenciju izvršenja.

Klauzula `ranked by` postoji u formalnoj gramatici i površinskom modelu, ali njena semantika još nije spuštена u izvršivi IR. Greška `ranking_semantics_deferred` zato nastaje tokom spuštanja u IR, a ne tokom parsiranja. Ovo razdvaja sintaktičku dostupnost od izvršne podrške: jezik može rezervisati stabilno mesto za budući koncept, ali ga ne sme označiti kao izvršiv dok nisu registrovani značenje skora i pravila rangiranja.

10.12 Površinski model, validacija i prevođenje u IR

Obrada DSL-a prolazi kroz više odvojenih provera. Sintaktička validacija odgovara na pitanje da li tekst pripada formalnoj gramatici i da li poštuje pravila rasporeda klauzula. Semantička validacija proverava ontološke pojmove, semantičke putanje, tipove referenci i kontekst klauzula. Provera softverskih mogućnosti proverava da li registrovani provajderi mogu da podrže tražene operacije i izvore. Tek nakon toga "spuštanje" tj transformacija proizvodi `WorkflowIR` i specifikaciju prikaza.

Ovi nivoi ne daju isti odgovor. Program može biti sintaktički ispravan, ali semantički neispravan zbog nepoznate putanje. Može biti semantički smislen, ali trenutno neizvršiv jer nijedna

registrovana softverska mogućnost ne podržava zahtevanu kombinaciju porekla, proizvoda i brokera. Može biti ispravan i imati podršku u deklarisanim softverskim mogućnostima, ali spuštanje može odbiti transformaciju ukoliko implementacija koraka nije završena.

Izveštaj softverskih mogućnosti koristi statusse `supported`, `unsupported`, `deferred` i `not_applicable`. `supported` označava dokazanu realizaciju. `unsupported` označava da tražena mogućnost nije nađena. `deferred` se koristi kada odluka zavisi od kasnije ili još neformalizovane softverske mogućnosti. `not_applicable` označava da se data provera ne odnosi na posmatranu klauzulu. Važno je da validacija mogućnosti ne poziva udaljene API-je; ona radi nad generisanim registrom.

Na primer, sledeći DSL:

```
object ZTF20aafqubg from ztf via antares
with lightcurve via antares
```

parser prvo predstavlja kao površinsku strukturu, tj AST SurfaceScript:

```
SurfaceScript(
  candidates=LookupCandidateSet(
    target_kind="object",
    ids=("ZTF20aafqubg",),
    origin="ztf",
    broker="antares",
    singular=True
  ),
  clauses=(
    RequirementClause(
      product="lightcurve",
      via="antares"
    ),
  )
)
```

Nakon validacije, izvodi se transformacija koju nazivamo spuštanje (engl. *lowering*) tj. funkcija `lower_surface()` prevodi površinsku strukturu u dva IR koraka:

```
LookupStep(
  target=TargetSelector(
    ids=["ZTF20aafqubg"],
    kind="object"
  ),
  sources=[
    Source(origin="ztf", broker="antares")
  ]
)
```

```
GetLightcurveStep(
    sources=[
        Source(origin="ztf", broker="antares")
    ]
)
```

Prvi semantički korak jeste LookupStep sa ciljem tipa object, a drugi GetLightcurveStep. U IR-u ostaju izvorna ograničenja origin=ztf i broker=antares, ali ne postoje URL, HTTP metoda niti fizičko ime parametra. Ta svojstva ulaze tek u EndpointPlan posle rada planera mogućnosti.

Spuštanje u IR takođe čuva razliku između udaljenog i lokalnog značenja. Npr, where se vezuje za početni korak pronalaženja i može kasnije biti podeljen na pushdown i rezidualni deo; filter postaje lokalni FilterStep; with lightcurve postaje semantički korak pribavljanja; order by odlazi u prikaz rezultata. Time jedan čitljiv program postaje skup precizno razdvojenih ugovora između DSL-a, orkestracije i integracionog sloja.

10.13 Inkrementalni nastavak aktivnog toka rada

DSL podržava i fragment koji nastavlja već izvršeni workflow. Prvi zahtev mora biti potpun program sa početnim skupom kandidata:

```
objects from ztf via lasair
inside (305.5822, -18.7909, 300arcsec)
with lightcurve via fink
```

Naredni zahtev može sadržati samo klauzule nad postojećim tokom:

```
filter detection@ztf:fink.quality.real_bogus >= 0.8
with lightcurve via lasair
```

Fragment se ne spaja sa starim tekstom. Parser proizvodi SurfaceFragment, a compile_surface_fragment ga kontekstualizuje neposredno prema kanonskom WorkflowIR aktivnog faznog izvršavanja. Novi koraci se dodaju na postojeću specifikaciju, dok se kompatibilan fizički prefiks ponovo koristi, a izvršavaju se samo novi koraci.

Ovo je važna granica pouzdanosti. Kada bi se nastavak zasnivao na konkatenaciji izvornog teksta, promena formatiranja, komentara ili buduće verzije gramatike mogla bi da promeni tumačenje već izvršenog dela. Oslanjanje na IR znači da je prethodni tok sačuvan u obliku koji orkestracija već razume i validira.

Fragment namerno ne dopušta inside, within, latest ili novi iskaz kandidata. Takva klauzula bi retroaktivno menjala početni skup, što nije nastavak postojećeg izvršenja već novi tok rada.

Dozvoljene su operacije koje rade nad postojećim materijalom: `filter`, `with`, `confirm`, `match`, `order by` i rezervisana `ranked by` klauzula.

10.14 Javni programski interfejs DSL sloja

Spoljni klijenti ulaze u DSL sloj preko dve stabilne funkcije: `validate_dsl` i `execute_dsl`. Time korisnički interfejs ne mora samostalno da povezuje parser, graf mogućnosti, planer, egzekuciju, normalizaciju i završne lokalne operacije.

```
from alertissimo.api import validate_dsl, execute_dsl

report = validate_dsl(source)

if report.is_runnable:
    execution = execute_dsl(source)
    payload = execution.to_dict()
```

`validate_dsl` ne kontaktira provajderske API-je. Funkcija obavlja parsiranje, ontološku validaciju, validaciju mogućnosti i kanonsko spuštanje u IR. Greške pri interaktivnom uređivanju DSL skripta vraćaju se kao podaci kroz `DSLValidationResult`, zajedno sa površinskim modelom, izveštajima i eventualnom kompilacijom.

Svojstvo `is_valid` znači da je izvor formalno parsiran i da je prošao semantičku validaciju.

`is_runnable` je stroži uslov: planiranje mogućnosti i spuštanje u IR moraju uspešno proizvesti `SurfaceCompilation`. Zbog toga sintaktički i semantički ispravan izraz može ostati neizvršiv kada je tražena mogućnost nepodržana, odložena ili je u pitanju rezervisana konstrukcija bez kanonskog IR značenja.

`execute_dsl` pokreće planiranje i stvarno izvršenje. Potpuni program započinje novi generički tok rada, dok fragment nastavlja aktivni. Povratni `DSLExecutionResult` objedinjuje izvor, površinski model, kompilaciju, fazno izvršenje i završni `WorkflowPortfolioResult`. Metode `to_dict` i `to_json` proizvode prikaz pogodan za veb prikaz sa metapodacima toka rada, tehničkim tokom izvršenja i kanonski serijalizovanim Portfolio rezultatima, bez izlaganja sirovih provajderskih izlaza kao primarnog javnog rezultata.

10.15 Celoviti primeri naučnih tokova

10.15.1 Višeopservaciono pronalaženje i poziciono povezivanje

Sledeći scenario formira zajednički početni skup iz LSST i ZTF porekla, a zatim povezuje kandidate prema položaju:

```
objects from lsst, ztf via alerce
inside (305.5822327501884, -18.7909207179724, 300arcsec)
match on position inside larcsec
```

Prvi radijus pripada pretrazi oblasti neba. Drugi je tolerancija pozicionog povezivanja tj. uparivanja. Rezultat MatchStep-a nije isto što i automatska harmonizacija zapisa istog identiteta u Portfolio; on predstavlja eksplicitnu naučnu operaciju povezivanja.

10.15.2 Obogaćivanje preko više brokera

```
objects from ztf via lasair
inside (305.5822327501884, -18.7909207179724, 300arcsec)
with lightcurve via fink
with lightcurve via lasair
order by summary.time.last_mjd desc
```

Isti semantički proizvod ovde se zahteva preko dva fizički različita izvora. Jedan provajder može imati eksplicitan lightcurve API poziv, dok drugi svetlosnu krivu fizički ostvaruje preko objektivne istorije ili niza alertova. DSL ne reprodukuje tu heterogenost. Graf mogućnosti i mapiranja objavljuju kako se svaka fizička realizacija povezuje sa kanonskim značenjem, a normalizacija konstruiše semantičke zapise sa očuvanom provenijencijom.

10.15.3 Lokalna odluka između udaljenih koraka

```
objects from ztf via lasair
inside (305.5822327501884, -18.7909207179724, 300arcsec)
with lightcurve via fink
filter detection@ztf:fink.quality.real_bogus >= 0.8
with lightcurve via lasair
```

Ovaj primer je arhitektonski važniji od prostog lanca API poziva. Posle Fink obogaćivanja rezultat se normalizuje i akumulira u Portfolio objektima. Lokalni FilterStep zatim dodatno izdvaja preostale kandidate. Orkestrator koristi njihove identifikatore i semantički materijal kao cilj sledećeg Lasair koraka obogaćivanja. Jezik tako ne opisuje samo redosled poziva već i tok podataka i odluka.

10.15.4 Potvrđivanje kvorumom

```
objects from ztf via alerce
inside (305.5822327501884, -18.7909207179724, 300arcsec)
latest 1
confirm by 2 via fink, alerce, antares
with lightcurve via fink
```

Kvorum dva od tri brokera je semantički deo ConfirmStep-a. Ovo je primer izvedenog koncepta koraka toka rada, koji ne postoji eksplicitno kao metoda u brokerskim interfejsima, a koji smo ustanovili analizom korisničkih potreba i primera.

10.16 DSL kao proverljiva meta za NLP i vizuelne interfejse

Fowler (2010) naglašava da DSL pokazuje svoju vrednost kada predstavlja stabilan jezik sporazumevanja između domenskog stručnjaka i izvršnog sistema. U Alertissimu formalna gramatika daje upravo takvu formalizovanu i proverljivu granicu za interfejs prirodnog jezika. LLM može da predloži skript, objasni ga ili ispravi na osnovu validacionog izveštaja, ali ne zaobilazi formalno jezgro. I vizuelni interfejs u kome bi se komponovali gradivni blokovi naučnog toka treba da proizvodi isti DSL ili istu proverenu površinsku strukturu. Time se ne formiraju dva nezavisna jezika - grafički i tekstualni - čija bi semantika vremenom mogla da se razide. Tekstualni editor, NLP prevodilac i vizuelni interfejs postaju različiti načini stvaranja istog formalnog artefakta.

Veza između slobodno formulisanog zahteva i formalnog skripta može se prikazati sledećim primerima. Tekst u navodnicima predstavlja mogući korisnički zahtev, a blok ispod njega odgovarajući DSL koji prolazi iste provere kao i ručno napisan skript.

„Pronađi pet najnovijih ZTF objekata u krugu od 300 lučnih sekundi oko zadatog položaja i preuzmi njihove svetlosne krive iz Finka.“

```
objects from ztf via lasair
inside (305.5822327501884, -18.7909207179724, 300arcsec)
latest 5
with lightcurve via fink
```

„Pronađi LSST objekte koje izabrani ALerCE klasifikator označava kao supernove sa verovatnoćom od najmanje 0,8 i dopuni ih svetlosnim krivama.“

```
objects from lsst via alerce
with classification from stamp_classifier_rubin_beta_20260421 via alerce where
best.class = "SN" and best.probability >= 0.8
with lightcurve via alerce
```

„Za poznati objekat ZTF20aafqubg preuzmi svetlosnu krivu iz Finka i kros-meč sa katalogom Gaia iz ANTARES-a.“

```
object ZTF20aafqubg from ztf via antares
with lightcurve via fink
with crossmatch from gaia via antares
```

„Od dva navedena ZTF objekta zadrži samo one čiji Fink podaci sadrže detekciju sa vrednošću real-bogus od najmanje 0,8, a zatim dodaj svetlosnu krivu iz Lasaira.“

```
objects ZTF20aafqubg, ZTF18acurdi from ztf via antares
with lightcurve via fink
filter detection@ztf:fink.quality.real_bogus >= 0.8
with lightcurve via lasair
```

„Pronađi LSST i ZTF objekte u istoj oblasti, poveži ih po položaju sa tolerancijom od jedne lučne sekunde i preuzmi njihove ALeRCE svetlosne krive.“

```
objects from lsst, ztf via alerce
inside (305.5822327501884, -18.7909207179724, 300arcsec)
match on position inside larcsec
with lightcurve via alerce
```

Šesti primer pokazuje razgovorni nastavak. Posle zahteva „Pronađi ZTF objekte u zadatoj oblasti i preuzmi njihove Fink svetlosne krive“ nastaje početni skript:

```
objects from ztf via lasair
inside (305.5822327501884, -18.7909207179724, 300arcsec)
with lightcurve via fink
```

Naknadni zahtev „Dodaj svetlosne krive iz Lasaira i poredaj rezultate prema vremenu poslednjeg posmatranja“ prevodi se u nastavak postojećeg toka:

```
with lightcurve via lasair
order by summary.time.last_mjd desc
```

Ova odluka je značajna i za reproducibilnost. Sačuvani DSL može da se verzioniše, citira uz rezultat i ponovo validira prema novom grafu mogućnosti. Prirodno-jezički razgovor bi tako ostao vrlo koristan alat, ali i dalje pomoćno sredstvo za formulisanje zahteva, dok formalni skript predstavlja proverljivu specifikaciju naučne namere.

10.17 Trenutni domet i ograničenja DSL sloja

Aktuelni DSL podržava formiranje kandidata semantičkom pretragom ili eksplicitnom object/alert identitetskom pretragom, ograničenja prostora/vremena/obima, semantičke predikate, lokalno

filtriranje, zahteve za proizvodima, potvrđivanje, povezivanje, uređivanje prikaza i inkrementalne fragmente. Implementacija je povezana sa realnim grafom softverskih mogućnosti, transformacijom u `WorkflowIR` i javnim funkcijama za validaciju i izvršenje.

Granice su namerno eksplicitne: `ranked by` je sintaktički rezervisan, ali nije izvršiv dok se ne formalizuju semantika evaluacije i brokerski ugovori. Kontinuirani monitoring alert toka nije deo sadašnje DSL gramatike niti postoji mehanizam izvršenja, iako `MonitorStep` postoji u širem IR rečniku. Vizuelni i NLP/LLM interfejsi predstavljaju buduće slojeve iznad formalnog jezika. Podrška za ugneždene SQL upite pojedinih brokera takođe je samo u planu.

Gramatika DSL-a namerno nije izvedena neposredno iz klasa orkestracionog IR-a. DSL predstavlja javni, korisnički sloj jezika, dok `WorkflowIR` predstavlja internu kanonsku reprezentaciju naučnog toka, pa između njihovih konstrukcija ne mora postojati odnos jedan-prema-jedan. Interna promena modela zato ne mora predstavljati promenu korisničkog jezika, niti svaki novi `Step` mora odmah dobiti posebnu površinsku sintaksu. Obrnuto, jezičke konstrukcije kao što je `order by` mogu predstavljati način izražavanja korisničke namere bez uvođenja zasebnog naučnog `Step` objekta. Veza između ova dva nivoa održava se eksplicitnim parserom i transformacijama u `WorkflowIR`, dok validacija i testovi proveravaju da podržane jezičke konstrukcije proizvode odgovarajuću kanonsku reprezentaciju. Ovakvo razdvajanje omogućava da se DSL razvija prema konceptima domena i načinu na koji ih korisnik izražava umesto prema unutrašnjoj organizaciji programskih klasa. Razdvajanje takođe omogućava da DSL verzija ostane stabilna kroz više iteracija sistema - stari skriptovi ostaju važeći čak i kada se interna implementacija pojedinih koraka promeni, sve dok se njihova semantička namera može mapirati na novi IR.

Aktuelna verzija DSL-a i `WorkflowIR` modela predstavlja tok kao uređenu sekvencu semantičkih koraka. Tok može biti zavisn od međurezultata: kandidati se mogu filtrirati, povezivati i koristiti kao ciljevi narednih operacija, pa se stvarni skup entiteta koji prolazi kroz tok dinamički menja tokom izvršenja. Međutim, trenutni model ne sadrži eksplicitne konstrukcije uslovnog grananja kojima bi se isti međurezultat razdvojio na više nezavisnih grana izvršenja. Scenario u kojem se, na primer, kandidati klasifikovani kao supernove prosleđuju spektroskopskom praćenju, a kandidati klasifikovani kao AGN zasebnoj radio analizi, zahtevao bi proširenje kontrolne strukture toka.

Ovo ograničenje ne svodi postojeći DSL na jezik upita: njime se već opisuju višestepeni naučni tokovi sa pribavljanjem podataka, lokalnom obradom, prenosom kandidata, akumulacijom

semantičkog materijala i zavisnostima od rezultata prethodnih koraka. Ipak, topologija jednog toka u aktuelnom prototipu ostaje linearna. Uvođenje uslovnih grana, kao i opštijih obrazaca grananja i spajanja tokova (engl. *fan-out*, *fan-in*) i paralelnog izvršenja, može predstavljati naredni korak razvoja orkestracionog modela.

11 Zaključak

11.1 Originalnost i doprinosi disertacije

Disertacija razmatra problem orkestracije heterogenih tokova podataka i softverskih servisa u astronomiji vremenskog domena sa stanovišta modelovanja informacionih sistema. Polazeći od okruženja u kojem različiti teleskopi, brokeri i servisi koriste sopstvene strukture i modele podataka, programske interfejse i načine izvršenja, istraživanje je usmereno na mogućnost da se naučni postupak opiše i izvršava nezavisno od tehnoloških razlika kao i da se razreše semantičke nedoumice. Rezultat je arhitektonski pristup u kojem su domensko značenje podataka i operacija, orkestracija naučnog toka i njegova konkretna fizička realizacija konceptualno razdvojeni, a zatim povezani formalizovanim mehanizmima prevođenja između odgovarajućih nivoa.

Jedan od centralnih elemenata predloženog pristupa predstavlja hijerarhijski semantički model astronomskih podataka. Model je projektovan tako da istovremeno ima praktičnu ulogu zajedničke, ponovo upotrebljive strukture za predstavljanje podataka iz različitih izvora i konceptualnu ulogu formalizovanja osnovnih pojmova domena i njihovih odnosa. Zahvaljujući toj dvostrukoj ulozi se podaci koji dolaze iz različitih brokera i servisa mogu predstaviti kroz iste domenske koncepte bez obzira na razlike njihovih izvornih šema. Tako semantička reprezentacija postaje stabilna tačka sistema na koju mogu da se oslone orkestracija, analiza i dalja obrada podataka.

Interoperabilnost sa konkretnim izvorima podataka ostvaruje se deklarativnim opisivanjem njihovih fizičkih krajnjih tačaka i mapiranjem između kanonskih semantičkih pojmova i struktura koje koriste pojedinačni programski interfejsi. Mehanizmi sistema na osnovu tih opisa obavljaju povezivanje naučne namere sa parametrima fizičkog poziva i transformaciju dobijenih rezultata u zajedničku reprezentaciju. Iz ovog pristupa proizilazi da se znanje o specifičnostima određenog programskog interfejsa / API-ja ne ugrađuje u centralnu orkestracionu logiku. Isti princip može se primeniti nezavisno od toga da li je fizički servis dostupan putem REST interfejsa, programske biblioteke ili nekog drugog mehanizma komunikacije, odnosno nezavisno od programskog jezika u kojem je konkretni servis realizovan.

Naučni tok se u predloženoj arhitekturi predstavlja pomoću semantičke međureprezentacije (IR) koja opisuje operacije u terminima njihovog značenja, a ne konkretnog načina izvršenja. Zahtev za pronalaženjem objekata, pribavljanjem svetlosne krive, klasifikacije ili dodatnih arhivskih podataka

predstavlja semantički korak naučnog postupka, dok se izbor servisa, konkretnog fizičkog poziva, parametara tog poziva i mesta izvršenja određuje tek tokom planiranja i izvršavanja toka. Time je omogućeno da naučni scenario ostane stabilan čak i kada se promeni njegova fizička realizacija. Istovremeno, eksplicitno predstavljanje koraka toka, njihovih zavisnosti, fizičkih izvršenja i porekla rezultata pruža osnovu za reproducibilnost i proverljivost izvršenih naučnih postupaka.

Najviši nivo interakcije sa ovim modelom predstavlja deklarativni domenski-specifični jezik. Njime se naučni zahtev izražava u terminima astronomskog domena, bez potrebe da korisnik unapred odredi način na koji će svaki njegov deo biti izvršen. Isti semantički iskaz može, u zavisnosti od raspoloživih servisa i podataka, biti realizovan udaljenim upitom, lokalnom obradom već pribavljenih podataka ili kombinacijom više fizičkih izvršenja. Nezavisnost jezika od konkretnog načina i mesta izvršenja predstavlja važnu posledicu celokupnog arhitektonskog razdvajanja: korisnički opis naučnog problema ne mora da prati promene u tehnološkoj infrastrukturi koja ga realizuje. Domenski-specifični jezik je svesno koncipiran kao deklarativan, dok njegova sintaksa - radi konceptualne jasnoće i lakšeg usvajanja - delimično prati obrasce poznate iz SQL-a. Za razliku od SQL-a, međutim, njegov predmet nisu relacione tabele i operacije nad njima, već opis naučne namere i kompozicija semantičkih koraka nad heterogenim izvorima podataka i servisima.

Predloženi pristup ne svodi orkestraciju na povezivanje API poziva u unapred određenom redosledu. Orkestracija se posmatra kao proces prevođenja između domenskog opisa naučnog problema i njegovih mogućih fizičkih realizacija, uz očuvanje značenja podataka, identiteta objekata, međurezultata i provenijencije. Takvo razdvajanje omogućava da se izvori podataka, servisi i implementacione tehnologije menjaju i proširuju uz minimalan uticaj na definiciju samog naučnog toka. Implementirani prototip pokazuje da je ovakav pristup primenljiv na postojeći ekosistem brokera LSST/ZTF astronomije vremenskog domena i da može da posluži kao osnova za dalje uključivanje novih izvora, analitičkih postupaka i načina korisničke interakcije.

Doprinosi disertacije mogu se, pored navedenih arhitektonskih i metodoloških rezultata, sagledati kroz naučnu, stručnu i širu društvenu dimenziju.

Naučni doprinos disertacije ogleda se u sistematizaciji problema orkestracije i semantičke interoperabilnosti u astronomiji vremenskog domena, kao i u razvoju i eksperimentalnoj proveri arhitektonskog pristupa koji razdvaja domensko značenje od konkretne tehnološke realizacije. Posebno se izdvajaju hijerarhijski semantički model koji istovremeno predstavlja ponovo

upotrebljivu strukturu podataka i ontološki, odnosno metamodelski opis ključnih koncepata domena; semantička međureprezentacija kojom se naučni postupak formalizuje kroz reproducibilne korake; deklarativna mapiranja i generički transformacioni mehanizmi kojima se semantički zahtevi povezuju sa heterogenim fizičkim API interfejsima nezavisno od programskog jezika i načina komunikacije; domenski-specifičan jezik kojim se naučna namera izražava nezavisno od načina i mesta izvršenja. Integracija ovih elemenata i njihova provera kroz prototip predstavljaju osnovu za dalju generalizaciju pristupa na druge naučne servise, izvore i tokove podataka. Važan projektantski izbor bio je da se ontologija i DSL ne tretiraju samo kao interna struktura i korisnička sintaksa već kao eksplicitne formalizacije domena namenjene i budućim mašinskim interfejsima, čime je postavljena osnova da NLP/LLM sloj prevodi prirodni jezik u proverljive semantičke reprezentacije, bez prenošenja odgovornosti za izvršnu semantiku na generativni model.

Stručni doprinos disertacije ogleda se u projektovanju i implementaciji prototipa Alertissimo kao proširive platforme za povezivanje postojećih astronomskih brokera i servisa u složene naučne tokove. Praktična vrednost rešenja je u tome što se specifičnosti pojedinačnih izvora izdvajaju u njihove deklarativne opise i mapiranja, dok zajednički mehanizmi obavljaju planiranje, vezivanje parametara, izvršavanje, normalizaciju i povezivanje međurezultata. Time se smanjuje potreba za ad hoc integracijama između svakog para alata, olakšava uključivanje novih izvora i sposobnosti, a očuvanje provenijencije i eksplicitnih koraka toka omogućava proveru, ponavljanje i dalje unapređivanje naučnih scenarija.

Društveni doprinos disertacije vezan je pre svega za jačanje naučne saradnje i dostupnosti velikih međunarodnih istraživačkih infrastruktura. Za naučnu zajednicu Srbije posebno je značajno da domaći istraživači aktivno učestvuju u razvoju takvih projekata kao što je LSST / Vera C. Rubin Observatory. Razvoj sopstvenih softverskih rešenja koja su deo širih softverskih (eko)sistema povećava sposobnost domaćih institucija da u takvim projektima učestvuju kao razvojni i naučni partneri, a ne samo kao korisnici podataka. U širem - globalnom - smislu, društveni doprinos rada može se odnositi na povezivanje globalno distribuiranih podataka i servisa, razmenu reproducibilnih naučnih tokova, kvantitativne i kvalitativne naučne pomake, otkrivanje i praćenje retkih i kratkotrajnih fenomena itd. Putem domenski-specifičnog, a kasnije i NLP/LLM interfejsa istovremeno će se smanjiti tehnička barijera za korišćenje savremenih naučnih resursa i podstaći prenos znanja.

11.2 Provera polaznih hipoteza

Rezultati istraživanja i implementacija prototipa omogućavaju neposrednu proveru osnovne hipoteze da se heterogeni tokovi podataka i specijalizovani softverski alati mogu orkestrirati prema potrebama naučnih scenarija u astronomiji vremenskog domena.

Realizovani sistem pokazuje da je moguće povezati više nezavisno razvijenih astronomskih brokera i servisa u jedinstven naučni tok, pri čemu pojedinačni koraci mogu koristiti različite izvore podataka, programske interfejsa i načine izvršenja. Posebno je pokazano da jedan fizički poziv može proizvesti više semantičkih objekata, da se podaci o istom objektu dobijeni iz različitih izvora mogu akumulirati u zajedničku reprezentaciju, kao i da se rezultati prethodnih koraka mogu koristiti za dinamičko određivanje ulaza narednih operacija. Podržane su posebne hipoteze koje se odnose na mogućnost formiranja opštih gradivnih blokova naučnog toka i na dinamičko uključivanje dodatnih izvora i operacija na osnovu međurezultata.

Rezultati takođe podržavaju hipotezu da je za ovakav oblik orkestracije potrebno uspostaviti zajednički semantički sloj između domenskih koncepata i konkretnih implementacionih interfejsa. Hijerarhijski semantički model omogućava da se heterogene strukture podataka različitih izvora predstave kroz zajedničke pojmove, dok deklarativna mapiranja i generički mehanizmi transformacije povezuju te pojmove sa konkretnim fizičkim reprezentacijama. Ovo omogućava separaciju orkestracione logike koja se može razvijati nezavisno od pojedinačnih API-ja i njihovih promena. Istovremeno, semantička međureprezentacija omogućava da se naučni tok modeluje kroz reproducibilne korake koji opisuju nameru operacije, dok se njihova konkretna fizička realizacija određuje odvojeno.

Potvrđena je i praktična opravdanost uvođenja domenski-specifičnog jezika za deklarativno definisanje naučnih scenarija. Hipoteza prema kojoj DSL treba da formalizuje domenske koncepte na višem nivou apstrakcije i smanji zavisnost naučnog opisa od konkretnih implementacija predstavlja jednu od centralnih ideja istraživanja.

Implementirani jezik omogućava da korisnik izrazi operacije nad astronomskim objektima i

podacima bez poznavanja detalja brokerskih API-ja, fizičkih naziva parametara ili transportnog mehanizma pojedinačnih servisa. Prevođenje takvog iskaza kroz semantičku međuprezentaciju, planiranje i izvršenje pokazuje da se deklarativni naučni opis može odvojiti i od načina izvršenja i od mesta izvršenja, što omogućava da se deo operacija realizuje na udaljenim servisima, a deo lokalno nad već pribavljenim i normalizovanim podacima.

Druga grupa hipoteza odnosi se na širi uticaj rastućeg obima astronomskih tokova na broj otkrića, razvoj istraživačkih metoda i multimodalnu karakterizaciju objekata. Rezultati ovog istraživanja podržavaju pretpostavku da povezivanje velikih, heterogenih i vremenski osetljivih skupova podataka, uz mogućnost njihovog orkestriranog i semantički usklađenog korišćenja, može predstavljati jedan od važnih činilaca budućih otkrića u astronomiji. Implementirani prototip pokazuje da je moguće objediniti više izvora i vrsta astronomskih podataka, povezati ih u zajedničke naučne tokove i dinamički uključivati dodatne servise i analitičke korake u zavisnosti od međurezultata, čime se stvaraju neposredni tehnički uslovi za efikasnije otkrivanje retkih i kratkotrajnih fenomena, bogatiju karakterizaciju objekata i razvoj složenijih naučnih scenarija. Dugoročna primena i povezivanje ovakvih i srodnih pristupa odvijaju se u širokom istraživačkom polju, u kojem naučni rezultati zavise od razvoja instrumenata, dostupnosti i kvaliteta podataka, izbora metoda, organizacije posmatranja i saradnje brojnih istraživačkih timova. Njihov naučni učinak zato prevazilazi okvir pojedinačne disertacije i zavisi od doprinosa više projekata i istraživačkih grupa. Potencijalni doprinos ove disertacije treba posmatrati kao deo šireg procesa izgradnje metodološke i softverske infrastrukture koja omogućava da se naučni potencijal novih generacija teleskopa, brokera, servisa i multimodalnih izvora podataka potpunije iskoristi.

11.3 Ograničenja istraživanja

Realizovano rešenje predstavlja funkcionalan istraživački prototip u poodmakloj fazi razvoja, sa implementiranim semantičkim modelom, deklarativnim mapiranjima, planiranjem i izvršavanjem radnih tokova, integracijom više nezavisnih astronomskih brokera i servisa, DSL-om i mehanizmima za očuvanje provenijencije. Predložena arhitektura nije proveravana samo na izolovanim primerima, već kroz niz međusobno različitih scenarija koji obuhvataju pretragu

objekata, pribavljanje i akumulaciju podataka iz više izvora, filtriranje, povezivanje objekata, različite oblike vezivanja ulaznih podataka i kombinovanje lokalnog i udaljenog izvršenja. Istraživanje je prevazišlo nivo konceptualnog ili pre-alfa prototipa i omogućilo praktičnu proveru ključnih arhitektonskih pretpostavki na stvarnim interfejsima i podacima postojećih astronomskih servisa. Najznačajnije preostalo ograničenje odnosi se zato pre na širinu empirijske validacije nego na osnovne funkcionalnosti sistema. Pri izboru slučajeva korišćenja nastojalo se da budu zastupljeni različiti tipovi podataka, provajdera i tokova izvršenja, ali astronomija vremenskog domena obuhvata izuzetno veliki prostor mogućih naučnih ciljeva. Predložena arhitektura biće najtemeljnije proverena kada je bude isprobao veći broj istraživača i istraživačkih grupa, a koji će formulisati scenarije proistekle iz sopstvenih naučnih pitanja, metodologija i potreba. Upravo takva upotreba - kroz različite oblasti astronomije, međunarodne kolaboracije i nezavisno osmišljene radne tokove - omogućiće da se identifikuju drugačiji ili ređi semantički obrasci, novi tipovi operacija i zahtevi koji nisu mogli biti unapred obuhvaćeni reprezentativnim skupom scenarija korišćenim u okviru jedne disertacije. Očekivane nadogradnje u fazi razvoja koja dolazi mogu biti ekstenzija gramatike DSL-a, uvođenje dodatnih koraka u međureprezentaciju tokova rada, kao i podešavanje i optimizacija interakcija sa operacijama brokera.

Važno ograničenje trenutne implementacije odnosi se na obradu kontinuiranih tokova događaja. Iako je podrška za monitoring u (skoro) realnom vremenu predviđena u semantičkom modelu orkestracije i odgovarajuća operacija je definisana kao poseban korak naučnog toka, u okviru realizovanog prototipa nije implementiran izvršni mehanizam za dugotrajno praćenje tokova, i reagovanje na novopristigle događaje. Evaluacija sistema je zato sprovedena nad realnim servisima i podacima, ali kroz vremenski ograničena i eksplicitno pokrenuta izvršenja, ne i kroz kontinuirano procesiranje živog astronomskog toka. Ovo ograničenje ne zahteva promenu osnovnog semantičkog ili orkestracionog modela. Naprotiv, eksplicitno predstavljanje monitoringa kao semantičke operacije omogućava da se buduća implementacija kontinuiranog izvršenja nadoveže na postojeće mehanizme planiranja, normalizacije, semantičkog povezivanja i provenijencije ali će predstavljati značajan i zahtevan razvojni i istraživački zadatak.

Ograničenje trenutne implementacije odnosi se i na širinu i opseg domena podataka, koji je u okviru disertacije svesno ograničen na ZTF i LSST, kako bi istraživanje moglo da bude metodološki zaokruženo i realizovano u realističnom vremenskom okviru. Multi-mesindžer izvori, poput gravitacionih talasa, neutrina, radio ili visokoenergetskih opservacija, nisu neposredno uključeni u

prototip, ali su njihovo uključivanje i odgovarajući tipovi podataka semantički skicirani i konceptualno obuhvaćeni arhitekturom. Ovakvo ograničenje je naročito opravdano imajući u vidu da i vodeći astronomski brokerski sistemi, čiji razvoj traje pet, deset, pa i više godina i u kojima učestvuju veliki međunarodni timovi, postepeno proširuju podršku na nove opservacione izvore i vrste podataka. Proširenje na multi-mesindžer astronomiju stoga predstavlja prirodan naredni korak razvoja, bez potrebe za promenom osnovnog arhitektonskog pristupa.

Postoji i čitav niz infrastrukturnih ograničenja softvera, vezanih za paralelizam i upravljanje otkazima, npr šta se dešava sa delimičnim realizacijama distribuiranih poziva, tajmoutovima, nedostupnim tačkama izvršenja itd. Pomeranje projekta u produkcionu fazu zahtevaće posvećen razvojni rad na ovom delu sistema.

Model provenijencije u prototipu je svesno ograničen na fundamentalni nivo koji obezbeđuje praćenje bez prekomernog usložnjavanja semantičkog modela. Ovakav kompromis je pre svega posledica prioriteta istraživanja i izabran je radi jednostavnosti modela, njegove ponovne upotrebljivosti i praktičnosti u orkestraciji naučnih tokova, uz očuvanje nivoa sledivosti dovoljnog za scenarija obuhvaćena istraživanjem. Ukoliko budući slučajevi upotrebe budu zahtevali finiju granularnost, semantički katalog se može proširiti, bez promene metamodela.

11.4 Pravci daljeg istraživanja i razvoja

Dalji razvoj treba prvenstveno usmeriti ka proširenju semantičkog modela i skupa podržanih naučnih operacija na osnovu konkretnih naučnih scenarija. Poseban istraživački pravac predstavlja odnos između tri semantičke reprezentacije koje sistem koristi - domenske ontologije, semantičke međureprezentacije i deklarativnog jezika - koje se delimično preklapaju, ali imaju različite uloge u predstavljanju podataka, naučnih operacija i korisničke namere. Njihovo formalnije međusobno povezivanje može doprineti daljoj generalizaciji arhitekture i preciznijoj validaciji naučnih scenarija.

Drugi pravac odnosi se na proširenje lokalnih semantičkih operacija koje se izvršavaju nad već normalizovanim podacima. Operacije povezivanja i filtriranja objekata predstavljaju početne primere takvog pristupa, dok naredni koraci mogu obuhvatiti poređenje različitih naučnih tvrdnji o istom objektu, izvođenje novih reprezentacija iz postojećih podataka, statističke agregacije i složenije analitičke postupke. Posebno je značajno istražiti u kojoj meri se granica između

udaljenog i lokalnog izvršenja može određivati dinamički, u zavisnosti od mogućnosti dostupnih servisa, troškova izvršenja i karakteristika konkretnog naučnog toka.

Konačno, prirodan sledeći korak predstavlja uvođenje AI interfejsa zasnovanih na obradi prirodnog jezika i velikim jezičkim modelima. DSL i ontologija su tokom razvoja namerno oblikovani kao formalna, mašinski čitljiva reprezentacija domena upravo zato da prirodni jezik ne mora neposredno da upravlja izvršenjem. LLM može da tumači slobodno formulisan naučni zahtev i prevede ga u DSL ili semantičku međureprezentaciju, nakon čega postojeći mehanizmi sintaksne i semantičke validacije, provere raspoloživih sposobnosti, planiranja i izvršenja ostaju formalno definisani i proverljivi.

Rezultat disertacije je konkretan prototip sistema, ali i osnova za dalja istraživanja načina na koji se semantika domena, deklarativno opisivanje naučne namere i fizička realizacija distribuiranih naučnih tokova mogu razvijati nezavisno, a ipak ostati formalno povezani u jedinstvenom softverskom sistemu.

12 Literatura

- Abbott, B. P., et al. [LIGO Scientific and Virgo Collaborations] (2016). Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.*, 116(6), 061102.
<https://doi.org/10.1103/PhysRevLett.116.061102>
- Aerts, C., Christensen-Dalsgaard, J. and Kurtz, D. W., 2010. *Asteroseismology*. Springer Science & Business Media. <http://doi.org/10.1007/978-1-4020-5803-5>
- Ahmed, A., Allen, J., Bhat, T., Burra, P., Fliege, C., Hart, S., Heldenbrand, J., Hudson, M., Istanto, D., Kalmbach, M., Kapraun, G., Kendig, K., Kendzior, M., Klee, E., Mattson, N., Ross, C., Sharif, S., Venkatakrishnan, R., Fadlelmola, F. and Mainzer, L. (2021). Design considerations for workflow management systems use in production genomics research and the clinic. *Scientific Reports*, 11.
<https://doi.org/10.1038/s41598-021-99288-8>
- Allen, J. F., 1983. Maintaining Knowledge about Temporal Intervals, *Communications of the ACM* 26(11): 832-843 <https://doi.org/10.1145/182.358434>
- Bassi, S., Sharma, K. and Gomekar, A., 2021. Classification of variable stars light curves using long short term memory network. *Frontiers in Astronomy and Space Sciences*, 8, p.718139.
<https://doi.org/10.3389/fspas.2021.718139>
- Bernstein, P. A. and Melnik, S. (2007). 'Model management 2.0: Manipulating richer mappings'. Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 1–12.
<https://doi.org/10.1145/1247480.1247482>
- Bishop, C. M., 2006. Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: *Springer*.
- Bloom, J. and Richards, J., 2011. Data mining and machine-learning in time-domain discovery & classification. *Advances in Machine Learning and Data Mining for Astronomy*, pp.89-112.
<https://doi.org/10.48550/arXiv.1104.3142>
- Bom, C.R., Mahabal, A., Bianco, F., Darc, P., Fraga, B., ..., Srećković, V.A, Vujčić V., ... & Variable Stars Science Collaboration. (2026). Toward decision-aware AI for LSST-scale time-domain astronomy. arXiv preprint arXiv:2606.05285. <https://doi.org/10.48550/arXiv.2606.05285>
- Borne, K., A machine learning classification broker for the LSST transient database. 2008, *Astronomische Nachrichten: Astronomical Notes*, 329, 255, <https://doi.org/10.1002/asna.200710946>
- Borucki, W. J., 2016. KEPLER Mission: development and overview. *Reports on Progress in Physics*, 79(3), p.036901. <https://doi.org/10.1088/0034-4885/79/3/036901>
- Budavári, T. and Szalay, A. S. (2008). 'Probabilistic cross-identification of astronomical sources'. *The Astrophysical Journal*, 679(1), pp. 301–309. <https://doi.org/10.1086/587156>
- Brahe, T. (1573) *De Nova et Nullius Aevi Memoria Prius Visa Stella* [Of the New and Never Previously Seen Star]. Hafniae (Copenhagen): Laurentius Benedicti.
- Cabrera-Vives, G., Moreno-Cartagena, D., Astorga, N., et al., ATAT: Astronomical Transformer for time series and Tabular data. 2024, *Astronomy and Astrophysics*, 689, A289, <https://doi.org/10.1051/0004-6361/202449475>
- Carrasco-Davis, R., Reyes, E., Valenzuela, C., et al., Alert Classification for the ALeRCE Broker System: The Real-time Stamp Classifier. 2021, *Astronomical Journal*, 162, 231, <https://doi.org/10.3847/1538-3881/ac0ef1>
- Catelan, M. and Smith, H. A., 2015. Pulsating stars. John Wiley & Sons.
- Cecconi, B., Louys, M., Derrière, S., Preite Martinez, A., Ochsenbein, F., Erard, S. and Demleitner, M., 2024. UCD1+ controlled vocabulary-Updated List of Terms Version 1.6. IVOA Endorsed Note 18 December 2024, p.1218.
- Chamberlin, D. D. and Boyce, R. F., 1974. SEQUEL: A structured English query language. Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access and Control, pp.249–264.
<https://doi.org/10.1145/800296.811515>
- Charnock, T., & Moss, A. (2017). Deep Recurrent Neural Networks for Supernovae Classification. *ApJL*, 837(2), L28. <https://doi.org/10.3847/2041-8213/aa603d>

- Coleman, T., Casanova, H., Pottier, L., Kaushik, M., Deelman, E. and da Silva, R. F., 2022. WfCommons: A framework for enabling scientific workflow research and development. *Future generation computer systems*, 128, pp.16-27.
- Debusscher, J., Sarro, L. M., Aerts, C., Cuypers, J., Vandenbussche, B., Garrido, R., & Solano, E. 2007, *A&A*, 475, 1159 <https://doi.org/10.1051/0004-6361:20077638>
- Deelman, E., Gannon, D., Shields, M. S., & Taylor, I. (2009). Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5), 528–540. <https://doi.org/10.1016/j.future.2008.06.012>
- Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., Ferreira da Silva, R., Livny, M. and Wenger, K. (2015). ‘Pegasus, a workflow management system for science automation’. *Future Generation Computer Systems*, 46, pp. 17–35. <https://doi.org/10.1016/j.future.2014.10.008>
- Demleitner, M., Dowler, P., Lusted, J., Mantelet, G., Morris, D., Nieto-Santisteban, M. A., Ohishi, M., O’Mullane, W., Ortiz, I., Osuna, P., Shirasaki, Y. and Szalay, A. S., 2023. Astronomical Data Query Language, Version 2.1. IVOA Recommendation, 15 December 2023. <https://www.ivoa.net/documents/ADQL/>
- Derriere, S., Gray, N., Mann, R., Preite Martinez, A., McDowell, J., McGlynn, T., Ochsenbein, F., Osuna, P., Rixon, G. and Williams, R. (2005). An IVOA Standard for Unified Content Descriptors, Version 1.1. IVOA Recommendation. <https://doi.org/10.5479/ADS/bib/2005ivoa.spec.0819D>
- Dewdney, P. E., Hall, P. J., Schilizzi, R. T., & Lazio, T. J. L. W. (2009). The Square Kilometre Array. *Proceedings of the IEEE*, 97(8), 1482–1496. <https://doi.org/10.1109/JPROC.2009.2021005>
- Di Stefano 2008 et al. New Opportunities In Microlensing And Mesolensing, American Astronomical Society Meeting Abstracts
- Di Stefano, R., 2008. Mesolensing explorations of nearby masses: from planets to black holes. *The Astrophysical Journal*, 684(1), pp.59-67. <https://doi.org/10.1086/528940>
- Dijkstra, E. W. (1982). Selected Writings on Computing: A Personal Perspective. New York: Springer-Verlag.
- Djorgovski, S. G., et al. (2001) 'The Digital Sky Survey and the Future of Astronomy', *Astrophysics and Space Science*, 276(2), pp. 1041–1054. <https://doi.org/10.1023/A:1017544026600>
- Dubath, P., et al. (2011). Random Forest Automated Supervised Classification of Hipparcos Periodic Variable Stars. *MNRAS*, 414(3), 2602–2617. <https://doi.org/10.1111/j.1365-2966.2011.18575.x>
- Chapter 5. EPL Reference: Clauses http://www.espertech.com/esper/release-5.2.0/esper-reference/html/epl_clauses.html
- Etzion, O., Niblett, P., 2010. Event Processing in Action, Manning, ISBN: 9781935182214
- Eyer, L. i Cuypers, J. (2000) 'Variable star statistics and the Hipparcos mission', u Szabados, L. i Kurtz, D. (ur.) The Impact of Large-Scale Surveys on Pulsating Star Research, ASP Conference Series, Vol. 203, pp. 71–73.
- Eyer, L., Mowlavi, N.: Variable stars across the HR diagram, 2007. arXiv:0712.3797v1. Astro-ph
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern recognition letters*, 27(8), pp.861-874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., et al., The Automatic Learning for the Rapid Classification of Events (ALeRCE) Alert Broker. 2021, *Astronomical Journal*, 161, 242, <https://doi.org/10.3847/1538-3881/abe9bc>
- Fowler, M. (2010). Domain-Specific Languages. Addison-Wesley Professional.
- Freire, J., Silva, C. T., Callahan, S. P., Santos, E., Scheidegger, C. E. and Vo, H. T. (2006). ‘Managing rapidly-evolving scientific workflows’. In: Provenance and Annotation of Data: International Provenance and Annotation Workshop, IPAW 2006, Lecture Notes in *Computer Science*, 4145, pp. 10–18. Springer. https://doi.org/10.1007/11890850_2
- Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., & Myers, J. (2007). Examining the challenges of scientific workflows. *Computer*, 40(12), 24–32. <https://doi.org/10.1109/MC.2007.421>
- Gothi, S. (2025). Event-Driven Microservices Architecture for Data Center Orchestration. *International Journal on Science and Technology*. <https://doi.org/10.71097/ijst.v16.i2.3113>.
- Graham, M. J., Djorgovski, S. G., Mahabal, A., Donalek, C., Drake, A. and Longo, G., 2012. Data challenges

- of time domain astronomy. *Distributed and Parallel Databases*, 30(5), pp.371-384.
- Graham, M., Kulkarni, S., Bellm, E., et al., The Zwicky Transient Facility: Science Objectives. 2019b, *Publications of the Astronomical Society of the Pacific*, 131, 078001, <https://doi.org/10.1088/1538-3873/ab006c>
- Graham, M. L., Bellm, E., Guy, L., Slater, C., Dubois-Felsmann, G. and Juric, M., 2024. LSST Alerts: Key Numbers. URL: <https://dmtn-102.lsst.io/> Vera C. Rubin Observatory Data Management Technical Note DMTN-102.
- Gray, J. (1986, January). Why do computers stop and what can be done about it?. In *Symposium on reliability in distributed software and database systems* (pp. 3-12).
- Gruber, T. R. (1993). 'A translation approach to portable ontology specifications'. *Knowledge Acquisition*, 5(2), pp. 199–220. <https://doi.org/10.1006/knac.1993.1008>
- Gupta, R., et al. (2025). Transfer Learning for Transient Classification: From Simulations to Real Data and ZTF to LSST. *MNRAS*, 542(1), L132. <https://doi.org/10.1093/mnras/slaf073>
- Harris, S. and Seaborne, A. (eds.), 2013. SPARQL 1.1 Query Language. W3C Recommendation, 21 March 2013. <https://www.w3.org/TR/sparql11-query/>
- Haswell, C. A., 2010. Transiting exoplanets. Cambridge University Press.
- Hellerstein, J. M. and Stonebraker, M. (1993). 'Predicate migration: Optimizing queries with expensive predicates'. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 267–276. <https://doi.org/10.1145/170035.170078>
- Hey, T., Tansley, S., & Tolle, K. (Eds.) (2009). The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, Redmond, Washington.
- Hoekstra, H., Yee, H. K. C. and Gladders, M. D., 2002. Current status of weak gravitational lensing. *New Astronomy Reviews*, 46(12), pp.767-781. [https://doi.org/10.1016/S1387-6473\(02\)00245-2](https://doi.org/10.1016/S1387-6473(02)00245-2)
- Iskandarli, L., Lintott, C. J., Croft, S., Stevance, H. and Weston, J., 2026. Anomaly Hunter for Alerts (AHA): Anomaly Detection in the ZTF Transient Alert Stream. arXiv preprint arXiv:2602.12955. <https://doi.org/10.48550/arXiv.2602.12955>
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al., LSST: From Science Drivers to Reference Design and Anticipated Data Products. 2019, *Astrophysical Journal*, 873, 111, <https://doi.org/10.3847/1538-4357/ab042c>
- Ivezić, Ž., Connolly, A.J. and Jurić, M., 2016. Everything we'd like to do with LSST data, but we don't know (yet) how. *Proceedings of the International Astronomical Union*, 12(S325), pp.93-102.
- Ivezić, Ž., Connolly, A. J., VanderPlas, J. T., & Gray, A. (2014). Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data. In *Statistics, data mining, and machine learning in astronomy*. Princeton University Press.
- Jegou du Laz, T., Coughlin, M. W., Bachant, P., Simones, J. E., Culino, T., Le Calloch, A., Sharma Chaudhary, S., Hall, X. J., Barna, T., Warshofsky, D. and Graham, M., 2025. BOOM and Babamul: a real-time, multi-survey, optical alert broker system operating at scale. arXiv e-prints, pp.arXiv-2511.
- Junell, A., Sasli, A., Nunes, F. F., Xu, M., Border, B., Rehemtulla, N., Rizhko, M., Qin, Y. J., Laz, T. J. D., Calloch, A. L. and Chaudhary, S. S., 2025. Applying multimodal learning to Classify transient Detections Early (AppleCiDER) I: Data set, methods, and infrastructure. arXiv preprint arXiv:2507.16088. <https://doi.org/10.48550/arXiv.2507.16088>
- Jurić, M., Kantor, J., Lim, K. T., Lupton, R. H., Dubois-Felsmann, G., Jenness, T., ... & Yoachim, P. (2015). The LSST data management system. *arXiv preprint arXiv:1512.07914*, <https://doi.org/10.48550/arXiv.1512.07914>
- Kim, D. W. and Bailer-Jones, C. A., 2016. A package for the automated classification of periodic variable stars. *Astronomy & Astrophysics*, 587, p.A18. <https://doi.org/10.1051/0004-6361/201527188>
- Kochanek, C. S., 2006, May. Part 2: Strong gravitational lensing. In *Saas-Fee Advanced Course* (Vol. 33, pp. 91-268).
- Kulkarni, S. R. (2012) 'The Cosmic Optical Transient Sky', *The Astrophysical Journal*, 103, pp. 243–257.
- Kundavaram, V. (2024). Event-Driven Data Pipelines : A Cloud-Based Approach to Real-Time Data Processing. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. <https://doi.org/10.32628/cseit24106183>.

- Lattner, C. and Adve, V. (2004). ‘LLVM: A compilation framework for lifelong program analysis & transformation’. International Symposium on Code Generation and Optimization, CGO 2004, pp. 75–86. <https://doi.org/10.1109/CGO.2004.1281665>
- Lemson, G., Laurino, O., Bourges, L., Cresitello-Dittmar, M., Demleitner, M., Donaldson, T., Dowler, P., Graham, M., Gray, N., Michel, L. and Salgado, J., 2018. VO-DML: a consistent modeling language for IVOA data models, Version 1.0. IVOA Recommendation. <https://doi.org/10.5479/ADS/bib/2018ivoa.spec.0910L>
- LSST Dark Energy Science Collaboration (LSST DESC) (bez datuma). The DESC ELAsTiCC Challenge. https://portal.nersc.gov/cfs/lsst/DESC_TD_PUBLIC/ELASTICC/ (pristupljeno 8. septembra 2026).
- LSST Science Collaboration (2009). LSST Science Book, Version 2.0. arXiv:0912.0201
- Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F. and Micol, A., 2017. Observation Data Model Core Components and its Implementation in the Table Access Protocol, Version 1.1. IVOA Recommendation. <https://doi.org/10.5479/ADS/bib/2017ivoa.spec.0509L>
- Luckham, D., 2011. Event Processing for Business: Organizing the Real-Time Enterprise, Wiley, ISBN: 9780470534854
- Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., & Zhao, Y. (2006). Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, 18(10), 1039–1065. <https://doi.org/10.1002/cpe.994>
- Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston: Prentice Hall.
- Matheson, T., Saha, A., Snodgrass, R., & Kececioglu, J., ANTARES: A Prototype Transient Broker System. 2014, in American Astronomical Society Meeting Abstracts, Vol. 223, American Astronomical Society Meeting Abstracts #223, 343.02
- Mattei, J. A. and Percy, J. R., 2001. Hands-On Astrophysics: variable stars for astronomy education and development. *Transactions of the International Astronomical Union*, 24(3), pp.88-94. <https://doi.org/10.1017/S0251107X0000047X>
- Meskhidze, H. (2024). Beyond Classification and Prediction: The Promise of Physics-Informed Machine Learning in Astronomy and Cosmology. In: Philosophy of Science for Machine Learning. Springer Nature. https://doi.org/10.1007/978-3-032-03083-2_18
- Mernik, M., Heering, J. and Sloane, A. M., 2005. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4), pp.316–344. <https://doi.org/10.1145/1118890.1118892>
- Mészáros, P., Fox, D. B., Hanna, C. and Murase, K., 2019. Multi-messenger astrophysics. *Nature Reviews Physics*, 1(10), pp.585-599. <https://doi.org/10.1038/s42254-019-0101-z>
- Moggi, E. (1991). Notions of computation and monads. *Information and Computation*, 93(1), 55-92.
- Michel, L., Cresitello-Dittmar, M., Bonnarel, F., Landais, G., Lemson, G., Louys, M. and Salgado, J., 2023. Model Instances in VOTables (MIVOT), Version 1.0. IVOA Recommendation. <https://www.ivoa.net/documents/MIVOT/>
- Möller, A., Peloton, J., Ishida, E. E., et al., FINK, a new generation of broker for the LSST community. 2021, *Monthly Notices of the Royal Astronomical Society*, 501, 3272, <https://doi.org/10.1093/mnras/staa3602>
- Möller, A., et al. (2024). Transient classifiers for Fink – Benchmarks for LSST. *A&A*, 691, A262. <https://doi.org/10.1051/0004-6361/202450370>
- Moreau, L. and Missier, P. (eds.) (2013). PROV-DM: The PROV Data Model. W3C Recommendation, 30 April 2013. <https://www.w3.org/TR/2013/REC-prov-dm-20130430/>
- Muthukrishna, D., et al. (2019). RAPID: Early Classification of Explosive Transients using Deep Learning. *PASP*, 131(1005), 118002. arXiv:1904.00014 <https://doi.org/10.1088/1538-3873/ab1609>
- Muthukrishna, D., et al. (2022). Real-time detection of anomalies in large-scale transient surveys. *MNRAS*, 517(1), 393–411. <https://doi.org/10.1093/mnras/stac2582>
- Narayan, G., et al. (2018). Machine-learning-based Brokers for Real-time Classification of the LSST Alert Stream. *ApJS*, 236(1), 9. arXiv:1801.07323 <https://doi.org/10.3847/1538-4365/aab781>
- Nordin, J., Brinnel, V., Van Santen, J., et al., Transient processing and analysis using AMPEL: alert management, photometry, and evaluation of light curves. 2019, *Astronomy & Astrophysics*, 631, A147, <https://doi.org/10.1051/0004-6361/201935634>
- North, J. (2008) Cosmos: An Illustrated History of Astronomy and Cosmology. Revised edn. Chicago:

University of Chicago Press.

- Object Management Group (2016). Meta Object Facility (MOF) Core Specification, Version 2.5.1. OMG. <https://www.omg.org/spec/MOF/2.5.1/>
- Odojin, O., Abayomi, A., Uzoka, A., Adekunle, B., Agboola, O. and Owoade, S. (2024). Designing Event-Driven Architecture for Financial Systems Using Kafka, Camunda BPM, and Process Engines. *International Journal of Scientific Research in Science, Engineering and Technology*. <https://doi.org/10.32628/ijrsrset25121178>.
- Paczynski, B., 1986. Gravitational microlensing by the galactic halo. *Astrophysical Journal*, Part 1 (ISSN 0004-637X), vol. 304, May 1, 1986, p. 1-5., 304, pp.1-5. <https://doi.org/10.1086/164140>
- Paczynski, B., 2000. Monitoring all sky for variability. *Publications of the Astronomical Society of the Pacific*, 112(776), pp.1281-1283. <https://doi.org/10.1086/316623>
- Paolucci, M., Kawamura, T., Payne, T. R. and Sycara, K. (2002). 'Semantic matching of web services capabilities'. In: The Semantic Web — ISWC 2002, Lecture Notes in Computer Science, 2342, pp. 333–347. Springer. https://doi.org/10.1007/3-540-48005-6_26
- Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12), 1053–1058. <https://doi.org/10.1145/361598.361623>
- Perez-Carrasco, M., Cabrera-Vives, G., Hernandez-García, L., et al., Alert classification for the alerce broker system: The anomaly detector. 2023, *The Astronomical Journal*, 166, 151, <https://doi.org/10.3847/1538-3881/ace0c1>
- Perlmutter, S., Aldering, G., Goldhaber, G., Knop, R. A., Nugent, P., Castro, P. G., Deustua, S., Fabbro, S., Goobar, A., Groom, D. E. and Hook, I. M., 1999. Measurements of Ω and Λ from 42 high-redshift supernovae. *The Astrophysical Journal*, 517(2), pp.565-586. <https://doi.org/10.1086/307221>
- “The” Natural History of Pliny: Transl. with Copious Notes and Illustrations. Vol. 2. Bohn, 1855.
- Prša, A., Batalha, N., Slawson, R. W., Doyle, L. R., Welsh, W. F., Orosz, J. A., Seager, S., Rucker, M., Mjaseth, K., Engle, S. G. and Conroy, K., 2011. Kepler eclipsing binary stars. I. Catalog and principal characterization of 1879 eclipsing binaries in the first data release. *The Astronomical Journal*, 141(3), p.83. <https://doi.org/10.1088/0004-6256/141/3/83>
- Richards, J. W., Starr, D. L., Butler, N. R., Bloom, J. S., Brewer, J. M., Crellin-Quick, A., Higgins, J., Kennedy, R., and Rischard, M. (2011). On machine-learned classification of variable stars with sparse and noisy time-series data. *ApJ arXiv:1101.1959*. <https://doi.org/10.1088/0004-637X/733/1/10>
- Sánchez-Sáez, P., Reyes, I., Valenzuela, C., et al., Alert Classification for the AleRCE Broker System: The Light Curve Classifier. 2021, *Astronomical Journal*, 161, 141, <https://doi.org/10.3847/1538-3881/abd5c1>
- Schmidt, B. P., et al. (1998) 'The High-Z Supernova Search: Measuring Cosmic Deceleration and Global Curvature of the Universe Using Type Ia Supernovae', *The Astrophysical Journal*, 507(1), pp. 46–63. <https://doi.org/10.1086/306308>
- Schneider, P., 2005, July. Gravitational lensing statistics. In Gravitational Lenses: Proceedings of a Conference Held in Hamburg, Germany 9–13 September 1991 (pp. 196-208). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Seaman, R., et al. (2011). Sky Event Reporting Metadata (VOEvent) Version 2.0. IVOA Recommendation. <https://www.ivoa.net/documents/VOEvent/>
- Servillat, M., Riebe, K., Boisson, C., Bonnarel, F., Galkin, A., Louys, M., Nullmeier, M., Renault-Tinacci, N., Sanguillon, M. and Streicher, O., 2020. IVOA Provenance Data Model, Version 1.0. IVOA Recommendation. <https://www.ivoa.net/documents/ProvenanceDM/>
- Shah, V., et al. (2025). ORACLE: A Real-Time, Hierarchical, Deep-Learning Photometric Classifier for the LSST. ArXiv:2501.01496 <https://doi.org/10.48550/arXiv.2501.01496>
- Shaw, M., & Garlan, D. (1996). Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall.
- Sigelman, B. H., Barroso, L. A., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., ... & Shanbhag, C. (2010). Dapper, a large-scale distributed systems tracing infrastructure.
- Singhal, P. (2024). Orchestration Workflows in Distributed Systems: A Systematic Analysis of Efficiency Optimization and Service Coordination. *International Journal For Multidisciplinary Research*. <https://doi.org/10.36948/ijfmr.2024.v06i06.30191>.

- Skoda, P. and Adam, F. eds., 2020. Knowledge discovery in big data from astronomy and earth observation: AstroGeoInformatics. Elsevier.
- Smith, K., Lasair: the transient alert broker for LSST: UK. 2019, The Extragalactic Explosive Universe: the New Era of Transient Surveys and Data-Driven Discovery, 51
- Smith, M. J., & Geach, J. E. (2023). Astronomia ex machina: a history, primer and outlook on neural networks in astronomy. *Royal Society Open Science*, 10(5). <https://doi.org/10.1098/rsos.221454>
- Sommerville, I. (2011). Software Engineering. 9th edn. Boston: Addison-Wesley.
- Stephenson, F. R. and Green, D. A., 2002. Historical supernovae and their remnants. Oxford University Press.
- Talaseela, R. (2025). Intelligent Workflow Orchestration for Enterprise Contexts. *European Journal of Computer Science and Information Technology*. <https://doi.org/10.37745/ejcsit.2013/vol13n272739>.
- Thoren, V. E. and Wilson, C., 1991. The lord of Uraniborg: a biography of Tycho Brahe.
- Tisserand, P., Le Guillou, L., Afonso, C., Albert, J. N., Andersen, J., Ansari, R., Aubourg, É., Bareyre, P., Beaulieu, J. P., Charlot, X. and Coutures, C., 2007. Limits on the Macho Content of the Galactic Halo from the EROS-2 Survey of the Magellanic Clouds. *Astronomy & Astrophysics*, 469(2), pp.387-404. <https://doi.org/10.1051/0004-6361:20066017>
- Tsimelzon, M. 2006, Complex Event Processing: Ten Design Patterns, [Online; accessed 12-2024]
- van der Walt, S. J., Crellin-Quick, A. and Bloom, J. S., 2019. SkyPortal: An astronomical data platform. *Journal of Open Source Software*, 4(37), p.1247. <https://doi.org/10.21105/joss.01247>
- van Deursen, A., Klint, P. and Visser, J., 2000. Domain-specific languages: an annotated bibliography. *ACM SIGPLAN Notices*, 35(6), pp.26–36. <https://doi.org/10.1145/352029.352035>
- Villar, V. A., et al. (2021). A Deep Learning Approach for Active Anomaly Detection of Extragalactic Transients. *ApJS*, 255(2). arXiv:2103.12102 <https://doi.org/10.3847/1538-4365/ac0893>
- Vujčić, V., Srećković, V. A., Babarogić, S. and Aleksić, J., 2025. An overview of astronomical transient brokers in Rubin era. *Contrib. Astron. Obs. Skalnate Pleso*, 55(2), pp.95-105. <https://doi.org/10.31577/caosp.2025.55.2.95>
- Vujčić, V., Srećković, V. A. and Babarogić, S., 2026. Alertissimo - a tool for orchestration of LSST broker streams. *Contributions of the Astronomical Observatory Skalnate Pleso*, 56(1), pp.186–193. <https://doi.org/10.31577/caosp.2026.56.1.186>
- Vujčić, V., Use of complex event processing engines in astronomy. 2014, in Big Data Conference proceedings, Szombately Hungary
- Vujčić, V. and Jevremović, D., 2020. Real-time stream processing in astronomy. In Knowledge Discovery in Big Data from Astronomy and Earth Observation (pp. 173-182). Elsevier. <https://doi.org/10.1016/B978-0-12-819154-5.00019-9>
- Vujčić, V. (2026). Alertissimo (Version 0.1.0) [Computer software]. Zenodo. <https://doi.org/10.5281/zenodo.22688389>
- W3C OWL Working Group, 2012. OWL 2 Web Ontology Language: Document Overview, Second Edition. W3C Recommendation. <https://www.w3.org/TR/owl2-overview/>
- W3C RDF Data Shapes Working Group, 2017. Shapes Constraint Language (SHACL). W3C Recommendation. <https://www.w3.org/TR/shacl/>
- Wadler, P. (1992). The essence of functional programming. In Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (pp. 1-14).
- Wang, J., Zheng, M. and Wei, T., 2026. COVENANT: Natural-Language Workflow Compilation for Aligned Agent Execution. arXiv preprint arXiv:2607.25400. <https://doi.org/10.48550/arXiv.2607.25400>
- Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer*, 25(3), 38–49. <https://doi.org/10.1109/2.121508>
- Wiener, N. (1948) Cybernetics: Or Control and Communication in the Animal and the Machine. New York: John Wiley & Sons.
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3, 160018. <https://doi.org/10.1038/sdata.2016.18>
- Williams, R. D., Djorgovski, S. G., Drake, A. J., Graham, M. J. and Mahabal, A., 2009. Skyalert: real-time astronomy for you and your robots. arXiv preprint arXiv:0906.2186. <https://doi.org/10.48550/arXiv.0906.2186>

- Williams, R. D., Francis, G. P., Lawrence, A., Sloan, T. M., Smartt, S. J., Smith, K. W. and Young, D. R., 2024. Enabling Science from the Rubin Alert Stream with Lasair. arXiv:2404.08315. <https://doi.org/10.48550/arXiv.2404.08315>
- Winn, J. N., 2010. Exoplanet transits and occultations (Vol. 55). Tucson, AZ: University of Arizona Press.
- Wood-Vasey, W. M., 2024. The Pitt-Google LSST Community Alert Broker. What Was That?-Planning ESO Follow up for Transients, Variables, and Solar System Objects in the Era of LSST, p.13.
- Young, D. R., 2023, Sherlock. Contextual classification of astronomical transient sources, <https://doi.org/10.5281/zenodo.8038057>
- Zadrozny, B. and Elkan, C., 2002, July. Transforming classifier scores into accurate multiclass probability estimates. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 694-699). <https://doi.org/10.1145/775047.775151>
- Zhang, G., Helfer, T., Gagliano, A. T., Mishra-Sharma, S. and Villar, V. A., 2024. Maven: a multimodal foundation model for supernova science. *Machine Learning: Science and Technology*, 5(4), p.045069. <https://doi.org/10.1088/2632-2153/ad990d>
- Zwicky, F. (1933) 'Die Rotverschiebung von extragalaktischen Nebeln', *Helvetica Physica Acta*, 6, pp. 110–127.

13 Biografija

13.1 Obrazovanje

Veljko Vujčić je rođen 1981. godine u Beogradu. Maturirao je 2000. godine u XXII beogradskoj gimnaziji a diplomirao je na Fakultetu organizacionih nauka (smer Informacioni sistemi) 2007. godine, kod prof Siniše Neškovića. Radio je u informatičkoj struci na izradi velikih poslovnih sistema i veb aplikacija. 2010. je upisao doktorske studije na FON-u, izborno područje Informacioni sistemi. Položio je sve ispite na doktorskim studijama, odbranio pristupni rad i 2016. prijavio tezu (kod prof Slađana Babarogića) "Otkrivanje i svrstavanje promenljivih astronomskih objekata obradom kompleksnih događaja u realnom vremenu", odobrenu od Naučno-nastavnog veća FON-a i Veća naučnih oblasti tehničkih nauka Univerziteta u Beogradu. 2018. godine privremeno je obustavio rad na doktoratu ali nastavlja da se bavi naučnoistraživačkim radom. 2025. godine ponovo je aktivirao status na doktorskim studijama.

13.2 Istraživački rad

Kandidat je zaposlen na Astronomskoj opservatoriji od februara 2011. Od 2011. do 2019, učestvovao je na projektu 44002 gde je bio najbliži saradnik rukovodioca projekta dr Darka Jevremovića. Posle završetka projektnog ciklusa učestvuje u radu grupe za Astroinformatiku. Naučnoistraživački rad vezan je za primenu informacionih tehnologija u astronomiji, fizici i srodnim poljima. Ostvaruje domaću saradnju sa Institutom za fiziku u Beogradu, na polju jonosferne i astrofizičke plazme, atomskih i molekulskih podataka i njihovog informacionog skladištenja i analize naročito kroz rad sa dr Vladimirom Srećkovićem i dr Bratislavom Marinkovićem. Naučni i stručni doprinos ostvaruje i kroz saradnju sa velikim međunarodnim projektima, poput LSST-a, gde je PI i rukovodilac grupe SER-STG i VAMDC-a. Oblast najrecentnijeg istraživanja je vezana za “brokere” – alate koji obrađuju astronomske tranzijente, specifično na konstruisanje domenskog jezika i “nad-brokera” za povezivanje funkcionalnosti iz heterogenih izvora podataka.

Učestvovao je kao član lokalnog organizacionog odbora u radu više međunarodnih

konferencija - SCSLSA, Big Data in Sky and Earth Observations, VAMDC workshop, NCAS, LSST at Europe, AsSpectro itd.

13.3 Naučna saradnja

Član je međunarodnih kolaboracija, učestvuje u radu nekoliko COST-akcija i bilateralnih projekata, i to:

- Rukovodilac i glavni istraživač (PI) srpske saradnje sa LSST/Rubin projektom SER-STG
- LSST-TVS - grupa za tranzijente i promenljivo nebo
- LSST Simulations Group - grupa za simulacije u okviru projekta, dok su se razvijali
- softverski moduli za simulirane podatke
- Rukovodilac SANU – BAN bilateralnog projekta “Posmatrački fenomeni i procesi u sistemima dvojnih zvezda”
- MC (član UO) COST akcije CA22133 PLANETS, član više radnih grupa u okviru
- nekoliko COST akcija i učesnik na više STSM misija
- Član astronomskog društva AstrO (Euroasian Astronomical Society)

<https://orcid.org/0000-0002-0525-1197>

13.4 Bibliografija

Ukupan broj bibliografskih jedinica kandidata je 53, od toga:

- jedan M21a rad,
- deset M21
- deset M22
- devet M23
- jedno poglavlje u monografiji

- četiri tehnička rešenja

Kandidat je kao prvi autor objavio četiri rada u kategoriji 21-23, i poglavlje u monografiji direktno vezano za temu disertacije.

Kandidat ima **1,044/1411/675/800** citata prema bazama ADS/Scholar/Scopus/WOS (pristupljeno 22. avgusta 2026). Glavna razlika u broju citata i Hirš indeksu uglavnom se vezuje za rad The LSST Data Management System koji je objavljen kao preprint, kao i za evidentiranost radova po bazama.

Spisak objavljenih naučnih radova

M13

1. **Vujčić, V.**, & Jevremović, D. (2020). Real-Time Stream Processing in Astronomy. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation* (pp. 173-182). Elsevier. ISBN: 978-0-128-19154-5. [10.1016/B978-0-12-819154-5.00019-9](https://doi.org/10.1016/B978-0-12-819154-5.00019-9)

M21a

1. Marinković, B. P., **Vujčić, V.**, Sushko, G., Vudragović, D., ... & Mason, N. J. (2015). Development of collisional data base for elementary processes of electron scattering by atoms and molecules. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 354, 90-95. [10.1016/j.nimb.2014.12.039](https://doi.org/10.1016/j.nimb.2014.12.039)

M21

1. Marinković, B.P., Sreckovic, V.A., Dujko, S., Tosic, S., Maljković, J.B., **Vujčić, V.** and Mason, N.J.J., 2025. Collisional ionization data for research of interstellar medium and planetary atmospheres. *Physica Scripta*. [10.1088/1402-4896/ade50b](https://doi.org/10.1088/1402-4896/ade50b)
2. Srećković, V.A., Ignjatović, L.M., Dimitrijević, M.S., **Vujčić, V.**, Tošić, S. and Iacob, F., 2025. Rydberg atoms and molecules in astrophysical and laboratory plasmas: Processes and data needed for modeling. *Advances in Space Research*. [10.1016/j.asr.2025.02.021](https://doi.org/10.1016/j.asr.2025.02.021)
3. Srećković, V.A., Delibašić-Marković, H., Ignjatović, L.M., Petrović, V. and **Vujčić, V.**, 2025. Photodissociation Processes Involving the SiH⁺ Molecular Ion: New Datasets for Modeling. *Data*, 10(11), p.185.
4. **Vujčić, V.**, Marinković, B.P., Srećković, V.A., Tošić, S., Jevremović, D., Ignjatović, L.M., Rabasović, M.S., Šević, D., Simonović, N. and Mason, N.J., 2023. Current stage and future development of Belgrade collisional and radiative databases/datasets of importance for molecular dynamics. *Physical Chemistry Chemical Physics*, 25(40), pp.26972-26985. [10.1039/D3CP03752E](https://doi.org/10.1039/D3CP03752E)
5. Srećković, V.A., Ignjatović, L.M., Dimitrijević, M.S., **Vujčić, V.**, Malović, M., Jevremović, D., Bezuglov, N.N. and Klyucharev, A.N., 2023. The Rydberg atom-atom collisions: Chemi-ionization cross-sections and rate coefficients in alkali-metal astrophysical and low-temperature laboratory plasmas. *Advances in Space Research*, 71(2), pp.1245-1251. [10.1016/j.asr.2022.04.069](https://doi.org/10.1016/j.asr.2022.04.069)
6. Srećković, V.A., Ignjatović, L.M., Kolarski, A., Mijić, Z.R., Dimitrijević, M.S. and **Vujčić, V.**, 2022. Data for Photodissociation of Some Small Molecular Ions Relevant for Astrochemistry and Laboratory Investigation. *Data*, 7(9), p.129. [10.3390/data7090129](https://doi.org/10.3390/data7090129)

7. Sreckovic, V.A., Šulic, D.M., **Vujčić, V.**, Mijic, Z.R. and Ignjatovic, L.M., 2021. Novel Modelling Approach for Obtaining the Parameters of Low Ionosphere under Extreme Radiation in X-Spectral Range. *Appl. Sci*, 11, p.11574. [10.3390/app112311574](https://doi.org/10.3390/app112311574)
8. Srećković, V.A., Šulić, D.M., Ignjatović, L. and **Vujčić, V.**, 2021. Low ionosphere under influence of strong solar radiation: diagnostics and modeling. *Applied Sciences*, 11(16), p.7194. [10.3390/app11167194](https://doi.org/10.3390/app11167194)
9. Dubernet, M. L., Antony, B. K., Ba, Y. A., Babikov, Y. L., ... , **Vujčić, V.**, ... & Zwölf, C. M. (2016). The virtual atomic and molecular data centre (VAMDC) consortium. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 49(7), 074003. [10.1088/0953-4075/49/7/074003](https://doi.org/10.1088/0953-4075/49/7/074003)
10. Di Valentino, E., Said, J.L., Riess, A., Pollo, A., .. , **Vujčić, V.**, ... & Saraf, C.S., 2025. The CosmoVerse White Paper: Addressing observational tensions in cosmology with systematics and fundamental physics. *Physics of the Dark Universe*, 49. [10.1016/j.dark.2025.101965](https://doi.org/10.1016/j.dark.2025.101965)

M22

1. **Vujčić, V.**, Marinković, B.P., Kopyra, J., Maljković, J.B., Srećković, V.A., Tošić, S., Aničić, N. and Mason, N.J., 2026. Data on Dissociative Electron Attachment Accommodated in the Structure of Belgrade Collisional Database ACol. *Atoms*, 14(7), p.52.
2. Tošić, S., Srećković, V.A. and **Vujčić, V.**, 2026. Reactive Collision Dynamics and Effective Cross-Sections in a Reduced-Dimensional Model Potential. *Atoms*, 14(8), p.68.
3. Delibašić-Marković, H., **Vujčić, V.**, Srećković, V.A. and Petrović, V., 2026. Analytical Study of Electron-Driven Ionization Dynamics and Plasma Formation in Intense Laser Fields. *Atoms*, 14(5), p.39.
4. Zamanov, R.K., Spassov, B., Konstantinova-Antova, R., Moyseev, M., Marti, J., Bode, M.F., **Vujčić, V.** and Sreckovic, V., 2025. Optical flickering in Mira and mass accretion rate onto the companion white dwarf. *New Astronomy*, p.102452.
5. Zamanov, R., Stoyanov, K.A., Latev, G., Marti, J., Takey, A., Elhosseiny, E.G., Christova, M.D., Minev, M., **Vujčić, V.**, Moyseev, M. and Marchev, V., 2024. Luminosity class of the symbiotic stars 4U1954+ 319 and ZZ CMi. *Serbian Astronomical Journal*, (208), pp.41-46. [10.2298/SAJ240206002Z](https://doi.org/10.2298/SAJ240206002Z)
6. Albert, D., Antony, B. K., Ba, Y. A., Babikov, Y. L., ... , **Vujčić, V.**, ... & Zwölf, C. M. (2020). A decade with VAMDC: Results and ambitions. *Atoms*, 8(4), 76. [10.3390/atoms8040076](https://doi.org/10.3390/atoms8040076)

7. Marinković, B. P., Srećković, V. A., **Vujčić, V.**, Ivanović, S., Uskoković, N., Nešić, M., ... & Mason, N. J. (2019). BEAMDB and MOLD—Databases at the Serbian Virtual Observatory for Collisional and Radiative Processes. *Atoms*, 7(1), 11. [10.3390/atoms7010011](https://doi.org/10.3390/atoms7010011)
8. Marinković, B. P., Jevremović, D., Srećković, V. A., **Vujčić, V.**, Ignjatović, L. M., Dimitrijević, M. S., & Mason, N. J. (2017). BEAMDB and MolD—databases for atomic and molecular collisional and radiative processes: Belgrade nodes of VAMDC. *The European Physical Journal D*, 71(6), 1-9. [10.1140/epjd/e2017-70814-6](https://doi.org/10.1140/epjd/e2017-70814-6)
9. Srećković, V. A., Ignjatović, L. M., Jevremović, D., **Vujčić, V.**, & Dimitrijević, M. S. (2017). Radiative and Collisional Molecular Data and Virtual Laboratory Astrophysics. *Atoms*, 5(3), 31. [10.3390/atoms5030031](https://doi.org/10.3390/atoms5030031)
10. Marinković, B. P., Bredehöft, J. H., **Vujčić, V.**, Jevremović, D., & Mason, N. J. (2017). Rosetta Mission: Electron Scattering Cross Sections—Data Needs and Coverage in BEAMDB Database. *Atoms*, 5(4), 46. [10.3390/atoms5040046](https://doi.org/10.3390/atoms5040046)

M23

1. **Vujčić, V.**, Srećković, V.A. and Babarogić, S., 2026. Alertissimo - a tool for orchestration of LSST broker streams. *Contributions of the Astronomical Observatory Skalnaté Pleso*, 56(1), pp.186–193. [10.31577/caosp.2026.56.1.186](https://doi.org/10.31577/caosp.2026.56.1.186)
2. **Vujčić, V.**, V. A. Sreckovic, S. Babarogic, and J. Aleksic. "An overview of astronomical transient brokers in Rubin era." *Contrib. Astron. Obs. Skalnaté Pleso* 55, no. 2 (2025): 95-105. [10.31577/caosp.2025.55.2.95](https://doi.org/10.31577/caosp.2025.55.2.95)
3. Sreckovic, V.A., Ignjatovic, L.M., Langovic, M. and **Vujcic, V.**, 2026. Molecular ion reaction rates for planetary atmospheres and the interstellar medium. *Contrib. Astron. Obs. Skalnaté Pleso*, 56(1), pp.140-148.
4. Sakan, N.M., Sreckovic, V.A. and **Vujcic, V.**, 2026. Advancing computational spectroscopy: machine learning approaches for reconstructing incomplete spectroscopic and collisional datasets. *Contrib. Astron. Obs. Skalnaté Pleso*, 56(1), pp.123-130.
5. Sreckovic, V.A., Marinkovic, B.P., Ignjatovic, L.M. and **Vujčić, V.**, 2025. MolD, EMol and ACol atomic and molecular databases for astrophysics: current stage and new directions of development. *Contrib. Astron. Obs. Skalnaté Pleso*, 55(2), pp.81-87. [10.31577/caosp.2025.55.2.81](https://doi.org/10.31577/caosp.2025.55.2.81)
6. Sreckovic, V.A., Ignjatovic, L.M., Tošić, S. and **Vujcic, V.**, 2023. The radiative processes involving some non-symmetric systems relevant for astrochemistry: data needed for modeling. *Contrib. Astron. Obs. Skalnaté Pleso*, 53(3), pp.107-114. [10.31577/caosp.2023.53.3.107](https://doi.org/10.31577/caosp.2023.53.3.107)

7. **Vujčić, V.**, Sreckovic, V.A. and Jevremovic, D., 2022. New database for collisional excitation/ionization processes of astrophysical interest. *Contrib. Astron. Obs. Skalnate Pleso*, 52(3), pp.71-77. [10.31577/caosp.2022.52.3.71](https://doi.org/10.31577/caosp.2022.52.3.71)
8. Jevremović, D., Srećković, V. A., Marinković, B. P., & **Vujčić, V.** (2020). Databases for collisional and radiative processes in small molecules needed for spectroscopy use in astrophysics. *Contributions of the Astronomical Observatory Skalnate Pleso*, 50(1), 44-54. [10.31577/caosp.2020.50.1.44](https://doi.org/10.31577/caosp.2020.50.1.44)
9. **Vujčić, V.**, Jevremović, D., Mihajlov, A. A., Ignjatović, L. M., Srećković, V. A., Dimitrijević, M. S., & Malović, M. (2015). MOL-D: A Collisional Database and Web Service within the Virtual Atomic and Molecular Data Center. *Journal of Astrophysics and Astronomy*, 36(4), 0. [10.1007/s12036-015-9344-y](https://doi.org/10.1007/s12036-015-9344-y)

M31

1. **Vujčić, V.** (2014). Use of complex event processing engines in astronomy. *New Challenges in Astro and Environmental Informatics in the Big Data Era, proceedings of the Big Data Conf*, pp 91-94. <https://www.gothard.hu/gao-mkk/memorabilia/bigdataconf-2014/proceedings/pdf/BigDataConf-proceedings.091-094.pdf>

M32

1. **Vujčić, V.** et al. (2015). Mol-D A Collisional Database Repository and Web Service Within the Virtual Atomic and Molecular Data Centre. *10th SCSLSA 06/2015*. ISBN 978-86-80019-70-3
2. **Vujčić, V.** (2016) APPLICATION OF CEP ENGINES IN ASTRONOMY. *LSST@Europe2 2016* pp 63. ISBN 978-86-80019-74-1
3. **Vujčić, V.** (2025). Alertissimo: building scientific workflows through broker orchestration. 4th
4. Regional LSST Workshop at HUN-REN CSFK Konkoly Observatory, Budapest, Hungary
5. https://events.konkoly.hu/RegionalLSSTWorkshop2025/documents/LSST_Regional_Bp2025_abstracts.pdf
6. **Vujčić, V.** (2025). Alertissimo: orchestrating LSST broker alerts for scientific workflows. Meeting on Operational and Research Capabilities for Better Understanding Solar-

Terrestrial Interactions, 09/25.

M33

1. Marinković, B. P, **Vujčić, V.**, Đorđević S., Ivanović S., Marinković B. P. , Jevremović D. & Mason N.J. (2014). Electron/Molecule Database Compatible with VAMDC Project. *Proc. 27th SPIG 2014, Contributed Papers & Abstracts of Invited Lectures, Topical Invited Lectures, Progress Reports and Workshop Lectures*, pp.42-45. ISBN: 978-86-7762-600-6.
2. Jurić, M., Kantor, J., Lim, K. T., Lupton, ... , **Vujčić, V.**, ... & Yoachim, P. (2015). The LSST data management system. *arXiv preprint* [arXiv:1512.07914](https://arxiv.org/abs/1512.07914)
3. Sreckovic, V. A., Mihajlov, A. A., Sakan, N. M., Ignjatovic, Lj. M., Jevremovic, D., Vujcic, V., Dimitrijevic, M. S. (2016). Astrophysical and laboratory plasmas: HF properties under extreme conditions. [arXiv:1612.04760](https://arxiv.org/abs/1612.04760)
4. Aleksić, J., **Vujčić, V.**, & Jevremović, D. (2016). Transient events in LSST survey data. *Proceedings of the International Astronomical Union*, 12(S325), 242-246. [10.1017/S1743921317000230](https://doi.org/10.1017/S1743921317000230)
5. Jevremović, D., **Vujčić, V.**, Srećković, V. A., Aleksić, J., Erkapić, S., & Milovanović, N. (2016). AlertSim-Serbian Contribution to the LSST. *Proceedings of the International Astronomical Union*, 12(S325), 122-125. [10.1017/S174392131601303X](https://doi.org/10.1017/S174392131601303X)
6. Aleksić, J., **Vujčić, V.**, & Jevremović, D. (2017). Alert simulator-a system for simulating detection of transient events on LSST. *Publications of the Astronomical Observatory of Belgrade*, vol. 96, pp. 331-335. [2017POBeo..96..331A](https://doi.org/10.1017/S1743921317000230)
7. **Vujčić, V.**, Aleksić, J., Neškovic, S., Jevremović, D. (2017). Use of complex event processing engines in time domain astronomy. *Publications of the Astronomical Observatory of Belgrade*, vol. 96, pp. 375-378. [2017POBeo..96..375V](https://doi.org/10.1017/S1743921317000230)
8. Srećković, V. A., Jevremović, D., **Vujčić, V.**, Ignjatović, L. M., Milovanović, N., Erkapić, S., & Dimitrijević, M. S. (2017). Mol-D a database and a web service within the Serbian virtual observatory and the virtual atomic and molecular data centre. *Proceedings of the International Astronomical Union*, 12(S325), 393-396. [10.1017/S1743921316012643](https://doi.org/10.1017/S1743921316012643)
9. Srećković, V. A., Mihajlov, A. A., Sakan, N. M., Ignjatović, L. M., Dimitrijević, M. S., Jevremović, D., & **Vujčić, V.** (2018). HF electric properties of the astrophysical plasmas. *Astronomical and Astrophysical Transactions*, 30(3), 307-314. [2018A&AT...30..307S](https://doi.org/10.1017/S1743921318000230)
10. Jevremović, D., **Vujčić, V.**, Srećković, V. A., Mihajlov, A. A., Ignjatović, L. M., & Dimitrijević, M. S. (2018). The MOL-D database VAMDC node for molecular collisional and radiative processes. *Astronomical and Astrophysical Transactions*, 30(3), 337-342.

11. Marinković, B.P., Srećković, V.A., Jevremović, D., **Vujčić, V.**, ... (2018). BEAMDB and MolD – Collisional and Radiative Databases at the Serbian Virtual Observatory, *Proc. 29th SPIG-2018, Contributed Papers & Abstracts of Invited Lectures, Topical Invited Lectures, Progress Reports and Workshop. Contributed Paper*, pp.23-26. ISBN 978-86-7306-146-7

M34

1. **VUJČIĆ, V.**, SREĆKOVIĆ, V.A., KOUNCHEV, O. and IACOB, F., 2024. A&M DATASETS FOR LTP TREATMENT OF PLANTS. *SPIG 2024*, p.158.
2. SREĆKOVIĆ, V.A., POP, N. and **VUJČIĆ, V.**, 2024. NEW MOLECULAR DATA FOR CONFINED MOLECULAR SYSTEMS AND ASTROCHEMICAL MODELLING. *SPIG 2024*, p.187.
3. TOŠIĆ, S., SREĆKOVIĆ, V. and **VUJČIĆ, V.**, 2024. SMALL MOLECULES ESSENTIAL TO ASTROPHYSICS: COLLISIONAL AND RADIATIVE PROCESSES. *SPIG 2024*, p.64.
4. SREĆKOVIĆ, V.A., POP, N., DIMITRIJEVIĆ, M.S., CHRISTOVA, M.D. and **VUJČIĆ, V.**, 2024. DATASET FOR PHOTODISSOCIATION OF SMALL MOLECULAR IONS. *SPIG 2024*, p.186.
5. **Vujčić, V.**, Aleksić, J., Nešković, S., Jevremović, D. (2014). Use of complex event processing engines in time domain astronomy. *XVII NCAS, book of abstracts*, p. 89.
6. Marinković, B.P., **Vujčić, V.**, Ivanović, S., Marinković D.B, Rixon, G., García, G., Jevremović, D., & Mason N.J. (2014). Representation of Electron and Positron Interactions with Biomolecules within RADAM Database. *Proc. 3rd Int. Conf. "Radiation Damage in Biomolecular Systems: Nanoscale Insights into Ion-Beam Cancer Therapy", Book of Abstracts*. p.106-107.
7. Jevremović D., Srećković V.A., Marinković B.P., **Vujčić V.**, Uskoković, N. & Ivanović, S. (2019). Belgrade nodes of vamdc: Databases for atomic and molecular collisional and radiative processes needed for spectroscopy, *Proc. XII SCSLSA 2019, Book of Abstracts*, p23. ISBN: 978-86-7589-134-5
8. Marinković, B. P., Srećković, V. A., **Vujčić, V.**, Ivanović, S., Uskoković, N., Nešić, M., ... & Mason, N. J. (2019). Cross Sections for Collisional and Radiative Processes: BEAMDB and MOLD Databases. *POSMOL 2019*, p. 126.
http://posmol2019.ipb.ac.rs/files/Book_POSMOL2019_Online.pdf. ISBN 978-86-7025-819-8

M51

1. Srećković, V. A., Šulić, D. M., **Vujić, V.**, Jevremović, D., & Vyklyuk, Y. (2017). The effects of solar activity: Electrons in the terrestrial lower ionosphere. *Journal of the Geographical Institute "Jovan Cvijic"*, SASA, 67(3), 221-233. [10.2298/IJGI1703221](https://doi.org/10.2298/IJGI1703221)

M83

1. Alert simulator for LSST (2015). **Вељко Вујчић**, Астрономска опсерваторија; Дарко Јевремовић, Астрономска опсерваторија; Јован Алексић, Астрономска опсерваторија; Миодраг Маловић, Иновациони центар Технолошко-металуршког факултета у Београду д.о.о.; Миодраг Маловић, Универзитет у Београду, Грађевински факултет; Сања Еркапић, Астрономска опсерваторија; RB 1590, MNTRS. <http://www.mpn.gov.rs/wp-content/uploads/2016/04/TEHNICKA-RESENJA-2011-2015-10-april.xls>

M84

1. Libraries being built in lib64 on OpenSUSE, when EUPS tables assume lib (2015). **Вељко Вујчић**, Астрономска опсерваторија; Дарко Јевремовић, Астрономска опсерваторија; Јован Алексић, Астрономска опсерваторија; Сања Еркапић, Астрономска опсерваторија; RB 1744, MNTRS. <http://www.mpn.gov.rs/wp-content/uploads/2016/04/TEHNICKA-RESENJA-2011-2015-10-april.xls>

M86

1. BEAMDB (2015). **Вељко Вујчић**, Астрономска опсерваторија; Владимир Срећковић, Универзитет у Београду, Институт за физику; Дарко Јевремовић, Астрономска опсерваторија; Сања Еркапић, Астрономска опсерваторија; RB 1612, MNTRS. <http://www.mpn.gov.rs/wp-content/uploads/2016/04/TEHNICKA-RESENJA-2011-2015-10-april.xls>
2. MOLD (2015). Анатолиј Михајлов, Универзитет у Београду, Институт за физику; **Вељко Вујчић**, Астрономска опсерваторија; Владимир Срећковић, Универзитет у Београду, Институт за физику; Дарко Јевремовић, Астрономска опсерваторија; Милан Димитријевић, Астрономска опсерваторија; Сања Еркапић, Астрономска опсерваторија; RB 1764, MNTRS. <http://www.mpn.gov.rs/wp-content/uploads/2016/04/TEHNICKA-RESENJA-2011-2015-10-april.xls>

ИЗЈАВА О АУТОРСТВУ

Име и презиме аутора _____ **Вељко Вујчић** _____

Број индекса _____ **2025/5039** _____

Изјављујем да је докторска дисертација под насловом

_____ **Оркестрација алата и хетерогених токова података** _____

_____ **у астрономији временског домена** _____

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

У Београду,

Потпис аутора

**ИЗЈАВА О ИСТОВЕТНОСТИ ШТАМПАНЕ И ЕЛЕКТРОНСКЕ ВЕРЗИЈЕ
ДОКТОРСКОГ РАДА**

Име и презиме аутора _____ **Вељко Вујчић** _____

Број индекса _____ **2025/5039** _____

Студијски програм _____ **Информациони системи и технологије** _____

Наслов рада _____ **Оркестрација алата и хетерогених токова података** _____
_____ **у астрономији временског домена** _____

Ментор _____ **проф др Слађан Бабарогић** _____

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањења у Дигиталном репозиторијуму Универзитета у Београду.

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум објаве рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

У Београду,

Потпис аутора

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

_____ **Оркестрација алата и хетерогених токова података** _____

_____ **у астрономији временског домена** _____

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

☐ 2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци. Кратак опис лиценци је саставни део ове изјаве).

У Београду,

Потпис аутора

1. **Ауторство.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.
2. **Ауторство – некомерцијално.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.
3. **Ауторство – некомерцијално – без прерада.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.
4. **Ауторство – некомерцијално – делити под истим условима.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.
5. **Ауторство – без прерада.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.
6. **Ауторство – делити под истим условима.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.