LLM-assisted Rule-based Policy Evaluation in Compute-to-Data Federated Workflows

N. Kokash, A. Belloum, P.Grosso

University of Amsterdam, Netherlands

Overview

Federated Data Processing

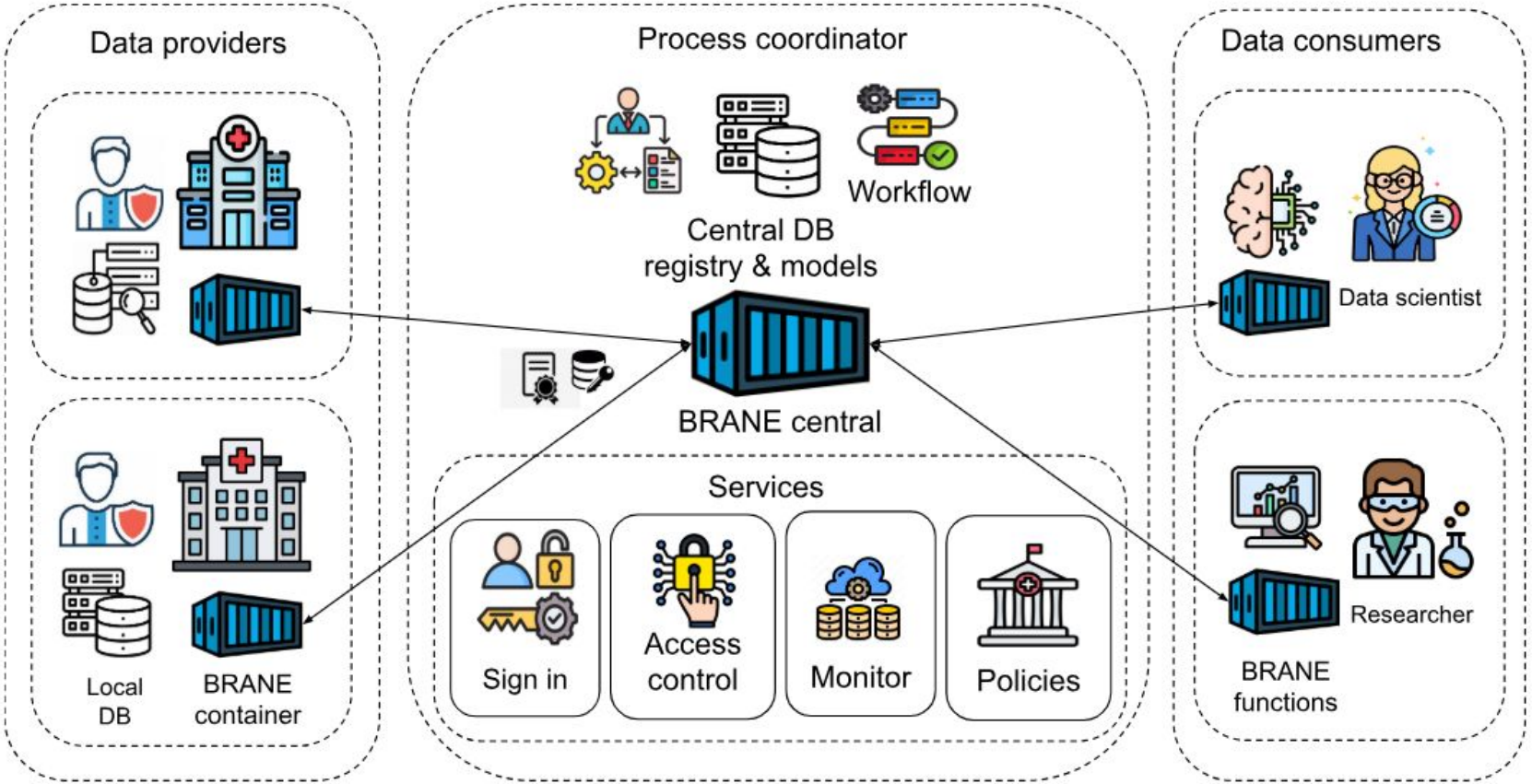
- Project needs vs available data matchmaking
- Experimenting with agentic workflows

Paper contributions

- Brane policy-reasoner extensions supporting OPA/Rego and LLM-based reasoners
- Comparative evaluation of eFLINT, OPA/Rego, and LLM reasoning for federated workflows
- Empirical evidence from TREC Clinical Trials: LLM-only agents are unreliable for eligibility decisions
- Hybrid policy architecture:
  LLM → fact extraction from unstructured criteria
  Rule engine → authoritative eligibility decision

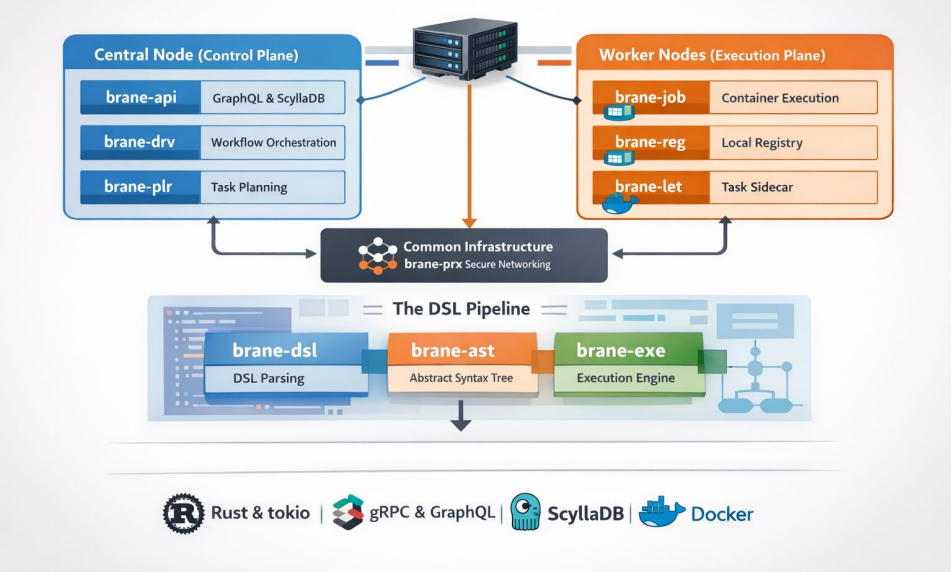
Project	Purpose / Contribution	Link
Brane Framework	Core privacy-focused application orchestration framework for distributed resources; containerized building blocks and programmable workflow orchestration	GitHub – BraneFramework
Extended Policy Reasoner	Policy reasoning with extended eFLINT + OPA/Rego support	GitHub
Matchmaking benchmark	Evaluation of LLM-based clinical-trial matchmaking	GitHub
Brane Easy Deployment	Simplified deployment and management of Brane infrastructure	GitHub

Brane for Federated Data Processing



Secure Data Exchange - Brane

- **Programmable Orchestration of Applications and Networking** (POAN) framework.
- Runs distributed applications across **geographically dispersed** IT resources
- Specifically targets environments with **sensitive data** (like the EPI project - Enabling Personalized Interventions)
- **Containerization** to encapsulate functionalities as portable building blocks
- DSL (BraneScript/Bakery) to allow **end-users** to compose workflows



Geo-Distributed Execution

Problem: Executing code on data that cannot be moved due to legal or technical reasons. **Solution**: The **Planner** (`brane-plr`) and **Worker nodes** allow **"computing to move to the data."** Brane handles the orchestration of containers across these sites transparently.

Security and Policy Enforcement

Problem: Ensuring that only authorized workflows can access sensitive datasets. **Solution**: Integration with a **Policy Reasoner** (`deliberation`) and the use of **Proxies** (`brane-prx`) to enforce network and data access rules at every step.

Ease of Use for Non-Programmers

Problem: Scientists often aren't distributed systems experts. **Solution**: **BraneScript** provides a high-level, Python-like syntax that hides the complexity of container management, networking, and data discovery.

Data-access policies

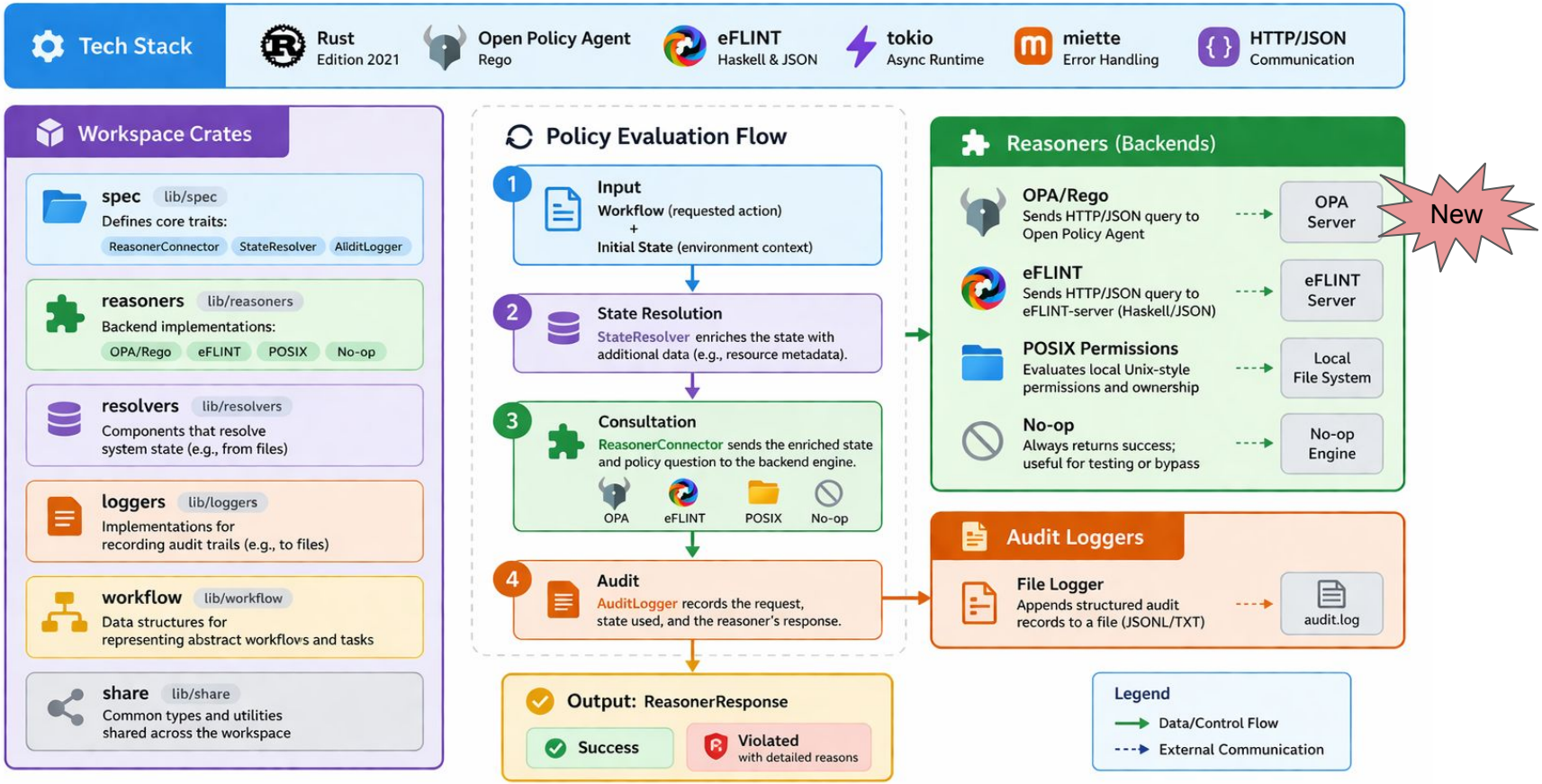
Control access: Who can access which data?
Constrain use: What processing is permitted?
Respect local rules: Each data provider can enforce its own policies.
Enable automation: Policies can be evaluated during workflow execution.
Ensure accountability: Decisions and actions are auditable.

A researcher may **access clinical data** if the researcher is **assigned** to the **approved project** and has completed the required **training**.

A data custodian may **release data to an approved workflow** only if the workflow is authorized, the requesting researcher is **permitted to access the data**, and the processing **purpose** is compatible with the approved project.

If a data subject has **withdrawn consent**, then the controller **must stop processing** the corresponding personal data **unless** another lawful basis for processing applies.

Policy reasoner



eFLINT vs Rego

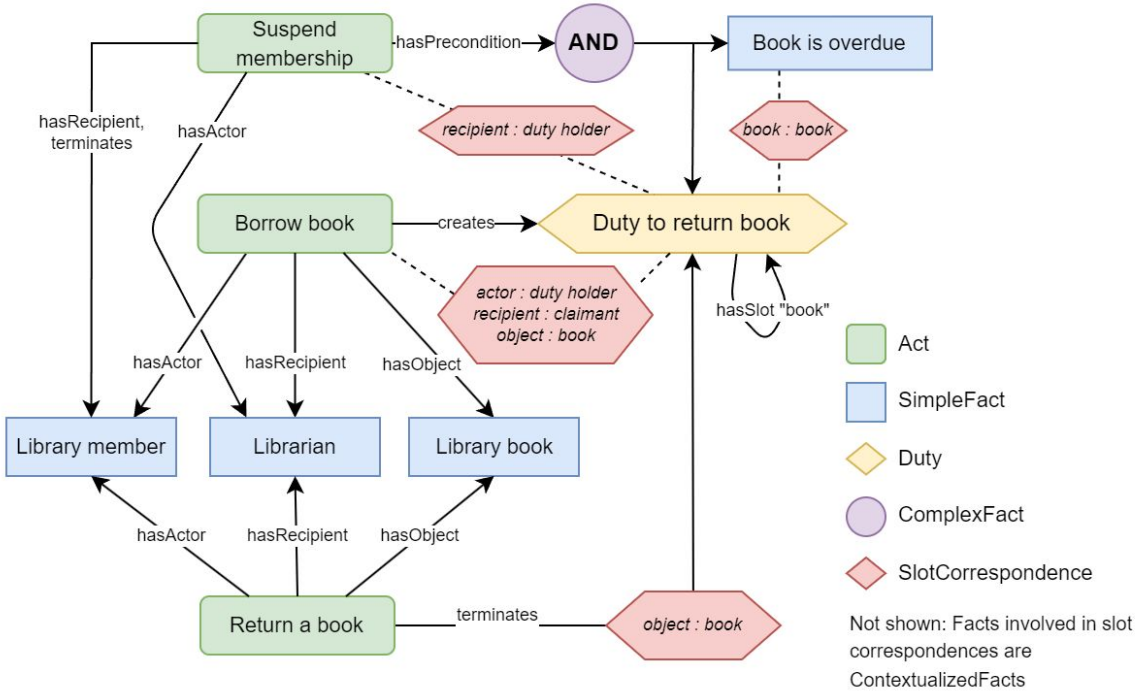
Aspect	eFLINT	OPA / Rego
Abstraction	Norms, duties, powers, violations	Rules over structured JSON
Temporal reasoning	Native fact lifecycle	Via input attributes
Legal traceability	**High**	Moderate
Performance	Interpreted; moderate	**Optimized; sub-ms**
Tooling	Research-grade (rule-editor,...)	**CLI, LSP** (Language Server Protocol)**, linter, CI/CD**
LLM generation	Moderate–low reliability	**High reliability**
Best use	Design-time normative audit	**Runtime enforcement**

Expressive, hard

Good for RBAC/ABAC, easy

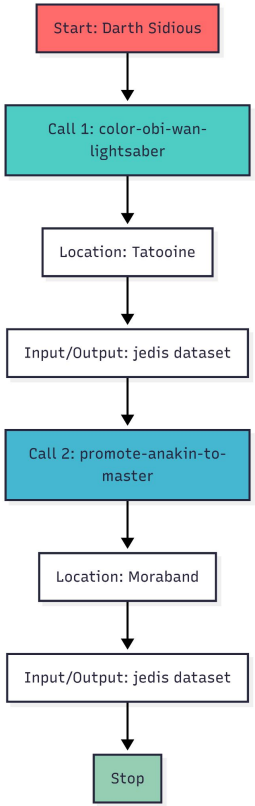
Flint / eFlint ontology example

1. Library members can borrow books.
2. The book should be returned within 28 days.
3. The membership of a library member that did not return a borrowed book in time, is suspended.



https://gitlab.com/normativesystems/knowledge-modeling/flint-ontology/

Role-based access control



Workflow

Step 1: Lightsaber Customization

- **Task**: `color-obi-wan-lightsaber`
- **Location**: Tatooine (the desert planet)
- **Operation**: Takes a "jedis" dataset as input and produces a modified "jedis" dataset as output
- **Purpose**: Appears to simulate customizing Obi-Wan Kenobi's lightsaber color

Step 2: Jedi Promotion

- **Task**: `promote-anakin-to-master`
- **Location**: Moraband (the ancient Sith homeworld)
- **Operation**: Takes the modified "jedis" dataset from Step 1 and produces a final "jedis" dataset
- **Purpose**: Simulates promoting Anakin Skywalker to Jedi Master rank
- **Completion**: The workflow ends after this step

Policy

Character Roles

- **Jedi Masters**: Yoda, Mace Windu, Obi-Wan Kenobi
- **Jedi Knights**: Anakin Skywalker, Kit Fisto, Plo Koon
- **Sith Lords**: Darth Sidious, Darth Vader, Darth Maul

Access Rules

- **Jedi Masters** can execute any task on any planet
- **Obi-Wan Kenobi** can specifically color his own lightsaber on Tatooine
- **Darth Sidious** can specifically promote Anakin to master on Moraband
- **Jedi Knights** can view but not modify the workflow
- **Sith Lords** can view but not modify the workflow (except for Darth Sidious's specific permission)
- All other characters are denied access by default

```
package jedis_policy

# Default deny all access
default allow = false

# Define roles for Star Wars characters
jedi_masters = {"Yoda", "Mace Windu", "Obi-Wan Kenobi"}
jedi_knights = {"Anakin Skywalker", "Kit Fisto", "Plo Koon"}
sith_lords = {"Darth Sidious", "Darth Vader", "Darth Maul"}

# Rule to allow Jedi Masters to perform any task on any planet
allow if {
    input.subject.id == jedi_masters[_]
    input.action == "execute"
}

# Rule to allow Obi-Wan to color his own lightsaber on Tatooine
allow if {
    input.subject.id == "Obi-Wan Kenobi"
    input.action == "execute"
    input.object.task == "color-obi-wan-lightsaber"
    input.object.location == "tatooine"
}

# Rule to allow only Darth Sidious to promote Anakin to master on Moraband
allow if {
    input.subject.id == "Darth Sidious"
    input.action == "execute"
    input.object.task == "promote-anakin-to-master"
    input.object.location == "moraband"
}

# Rule to allow Jedi Knights to view but not modify the workflow
allow if {
    input.subject.id == jedi_knights[_]
    input.action == "view"
}

# Rule to allow Sith Lords to view and execute the workflow
allow if {
    input.subject.id == sith_lords[_]
    input.action == "view"
}
```

Policy specification languages

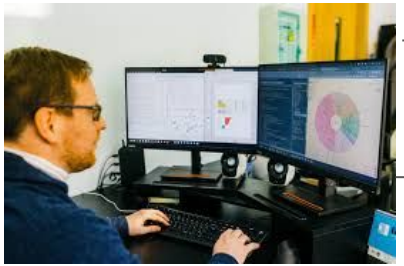
Languages for defining access control and data privacy policies.

- **[+] Rego (Open Policy Agent, OPA)** – A declarative policy language used for expressing fine-grained access control policies, especially in cloud and microservices environments.
- **XACML (eXtensible Access Control Markup Language)** – A widely used XML-based standard for attribute-based access control (ABAC).
- **AIR (Accountability In RDF)** – A rule-based, RDF-based policy language designed for expressing privacy policies in the semantic web.
- **PSL (Privacy Specification Language)** – A formal language designed to express privacy preferences and data handling requirements.
- **PawS (Privacy-Aware Security Language)** – A semantic-web-based language for privacy policy enforcement.
- **Legalease** – A policy language developed by Microsoft Research for expressing compliance policies in natural-language-like syntax.

Languages used in privacy-preserving computing and federated learning.

- **TLA+ (Temporal Logic of Actions)** – Used in formal verification of privacy-sensitive access control policies.
- **PPL (PrimeLife Policy Language)** – A privacy policy language developed for user-centric identity management.
- **LegalRuleML** – An XML-based language used to model legal norms, including privacy policies.
- **AAL (Attribute-based Access Language)** – Used in federated identity management and access control frameworks.

Federated Data Processing



Research institute

Can I have your data?

No!

But I will make your life better!

Who are you? What data do you want?

How do I know you are who you say you are?
Where are you from?
What are you doing?
What will I get from it?
Are you going to share data with someone else?
...

Medical center

- **EHDEN – European Health Data & Evidence Network**
  A large-scale European federated network for real-world healthcare data. Data remain at the local institutions, but are harmonised to the **OMOP Common Data Model**, enabling distributed epidemiological, drug-safety and effectiveness studies. The network has 100+ data partners across 25+ countries, with more than 291 million people represented.

- **DataSHIELD / DARAH oncology network**
  Federated analysis infrastructure in which researchers send statistical analyses to participating sites rather than receiving patient-level data. The DARAH project deployed it across three French hospitals (Toulouse, Reims and Foch) for an oncology study; all planned aggregate analyses were successfully performed without centralising patient data. It also demonstrated integration of differential privacy.

Practical problems

Researcher

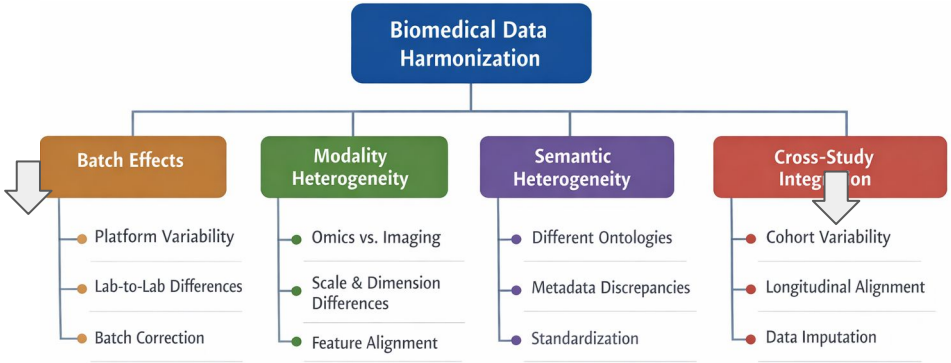
- How do I process data I cannot see?
- What do I do if these data are not in the format I need?
- What if the data is not compatible / useful for our needs?
- …

- Data agnostic ML
- Collect metadata (+ synthetic data) and adapt processing to a specific provider

Data provider

- How do I enable research without jeopardising my data?
- They use formats and DSLs I do not understand or have no resources to comply with…
- …

- Clear instructions + tool helpers
- Framework-level protection and automation



Onboarding policies

- Not all data are useful
- Templates / schema to describe goal and data
- Use project / data descriptions to decide proceed or not

Health DCAT-AP

DCAT-AP — a machine-readable description that enables discovery across organisations.

```
input_data = {
    "location": location,
    "acceptable_locations": acceptable_locations
}

prompt = f"""
You are a policy evaluation engine. Decide whether to accept a participant
    based on the following input data: {json.dumps(input_data, indent=2)}

Rule: Accept a participant (allow: true) if their location is a valid
    representation or sub-region of one of the acceptable_locations.
    Otherwise, deny (allow: false).

Return ONLY a JSON object in the following format:
{{"allow": true}} or {{"allow": false}}
"""
```

Rego / Open Policy Agent (OPA)

- Policy-as-Code framework for declarative policy definition
- Decouples policy decisions from application logic
- Enables fine-grained authorization and compliance enforcement
- Widely used for cloud-native infrastructure, APIs, and data governance

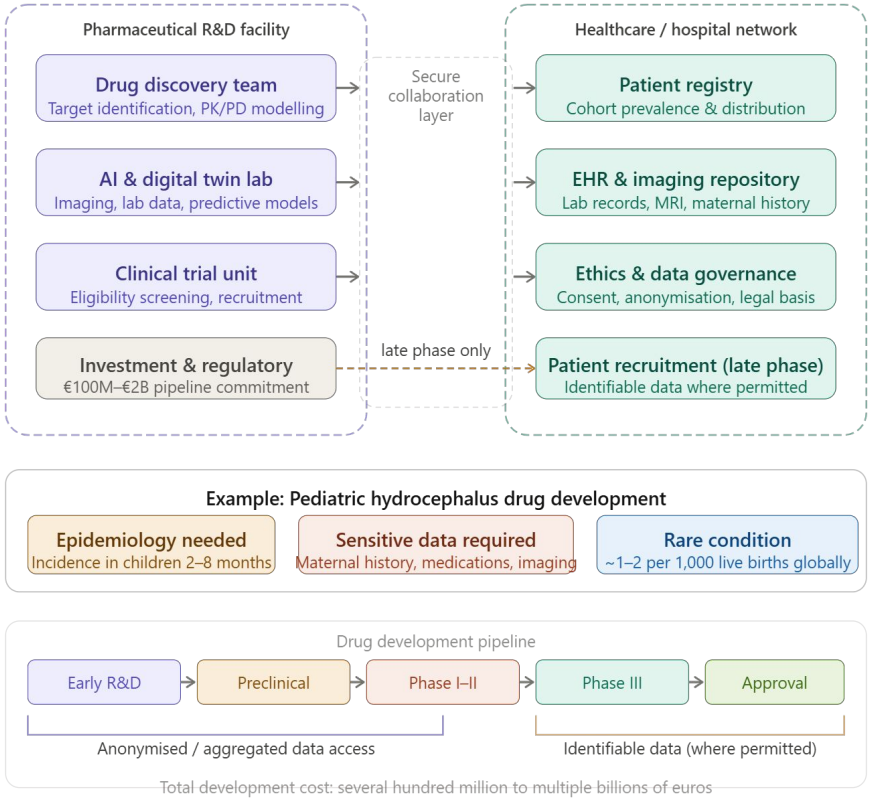
Typical uses: Access control (RBAC / ABAC), API authorization, Compliance and security policies

```
# Expected input shape: {
# "location": "EU", "acceptable_locations": ["EU", "NL", "BE", "DE"...] }

default allow = false

allow if {
  # Ensure the location is present in the acceptable_locations collection
  input.acceptable_locations[_] == input.location
}

# Returns: {"allow": true|false}
decision := {"allow": allow}
```

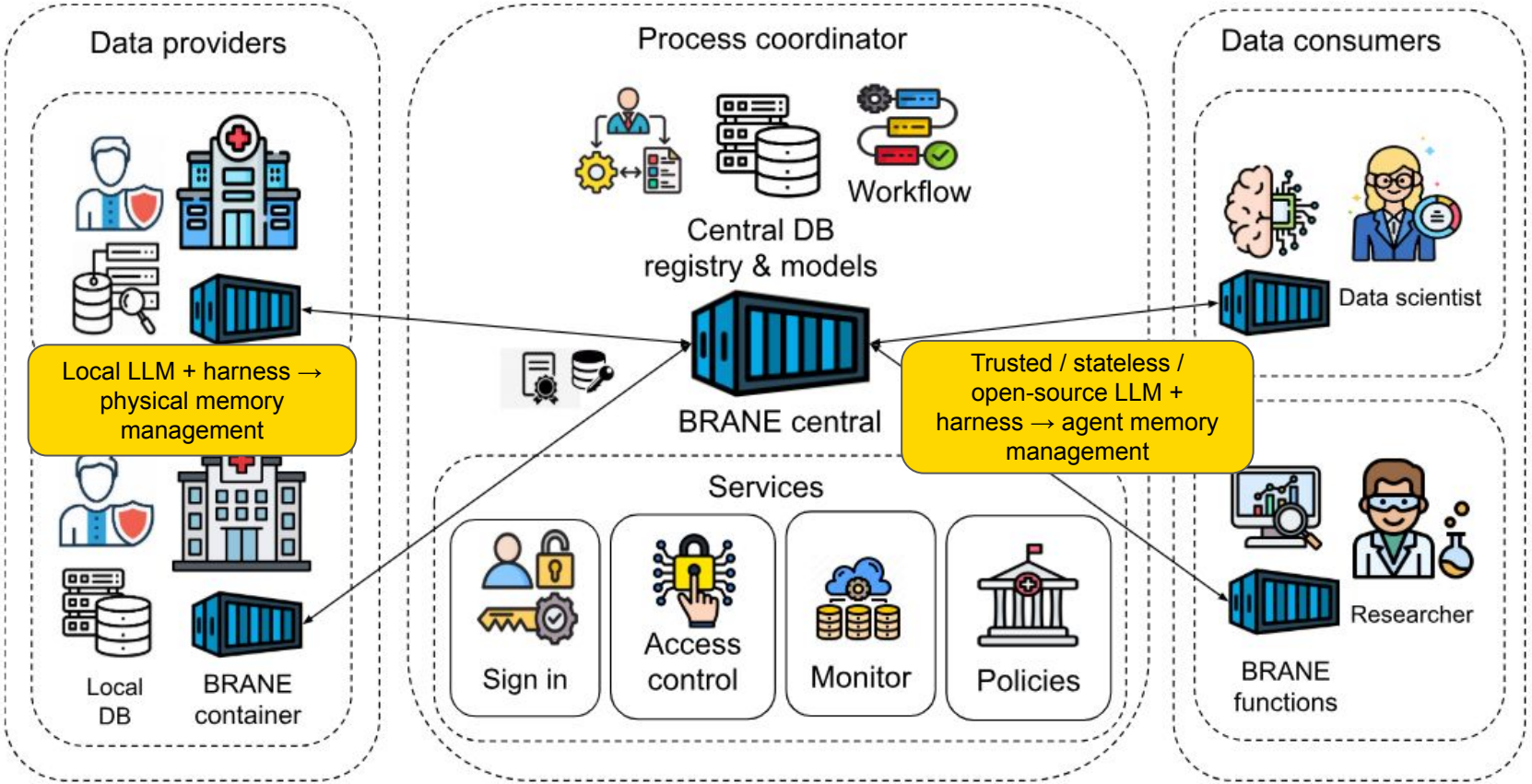
Medical / healthcare use cases

Global research and Regulatory Surveillance

1. **Clinical trial patient selection.** An R&D facility gathers information about cohorts of patients, such as the frequency of condition occurrence and circumstances in which the condition developed:
    - selected patients' medical history, laboratory tests, and images.
    - personal data to identify and recruit patients for clinical trials.
2. **Nationwide study of effects of a screening program**
    - e.g., colonoscopy from age of 35 vs from age 50
    - does it help early cancer detection - count cancer discoveries in asymptomatic patients in the age group 35 vs 50, average tumor size/staging, treatment outcomes...
3. **Hospital quality controls**: count number of in-hospital mortality and 30-day readmission cases…

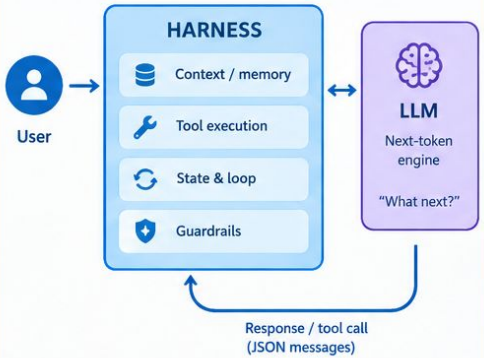


Federated agents in Brane



1. What is an Agent, Physically?

The LLM is the brain. The harness is the body.



Physically, the harness is just a program.

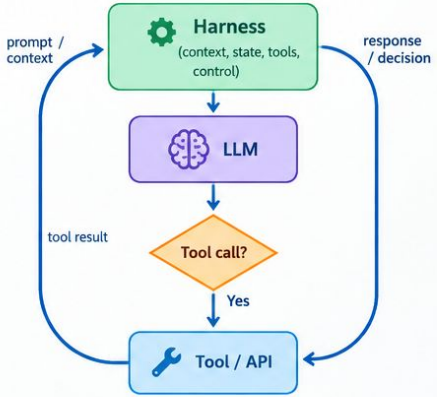
- Calls the LLM API
- Maintains the conversation/context array
- Interprets model output
- Executes tools/functions
- Feeds results back to the LLM
- Repeats until the task is complete

LLM API is stateless; the harness builds, maintains and protects the state around it.

2. The Agent Loop

Where the "Agent" Actually Lives

An agent emerges from a simple control loop.



Each capability is a patch around the basic loop:

Problem	Harness capability
LLM cannot affect world	Tools
One action is insufficient	Agent loop
Context becomes expensive	Caching
Context window fills	Compaction / context management
One context cannot handle task	Subagents
Agent drifts	Steering / control
Agent does something unsafe	Guardrails

3. What You Actually Build

Agent development = engineering the harness around the model.

The engineering challenge is therefore not only "make the LLM smarter."

It is to build the body around the LLM:

Context → Actions → State → Feedback → Control → Safety

Agent frameworks, MCP, RAG, skills, hooks, subagents and guardrails are extensions of the harness, not fundamentally different kinds of AI.

Matchmaking agent (finds suitable data records)

Inclusion • ECOG performance status **0–1** • Life expectancy **≥12 weeks** • Adequate bone marrow: ANC **≥1.5 × 10⁹/L**, platelets **≥90 × 10⁹/L**, Hb **≥9 g/dL** • Adequate hepatic & renal function • LVEF **≥50%**

Exclusion • Known/suspected **autoimmune disease** • Systemic corticosteroids within **28 days** • Prior **LAG-3 antibody** treatment • **Untreated CNS metastases** • Treated, stable CNS metastases permitted after **1-month washout**

eFLINT

```
Fact ecog-eligible Identified by patient
  Holds when ecog-score(patient) = 0
         Or  ecog-score(patient) = 1.

Fact adequate-life-expectancy Identified by patient
  Holds when life-expectancy-weeks(patient) >= 12.

Fact adequate-bone-marrow Identified by patient
  Holds when adequate-anc(patient)
         And adequate-platelets(patient)
         And adequate-haemoglobin(patient)
         And no-recent-blood-products(patient)
         And no-recent-growth-factors(patient).

Fact adequate-cardiac-function Identified by patient
  Holds when lvef-percent(patient) >= 50.

Fact has-autoimmune-disease Identified by patient.

Fact recent-immunosuppression Identified by patient
  Holds when
    days-since-last-immunosuppressant(patient) <= 28.

Fact permissible-brain-metastases
  Identified by patient
  Holds when Not has-brain-metastases(patient)
         Or (stable-brain-metastases(patient)
             And steroids-discontinued-for-brain-mets(
               patient)).
```

Rego

```
reasons contains "I-4: ECOG must be 0 or 1" if {
    not _ecog_eligible
}
reasons contains
  "I-5: Life expectancy < 12 weeks" if {
    input.patient.prognosis
      .life_expectancy_weeks < 12
}
reasons contains "I-6: ANC < 1.5e9/L" if {
    input.patient.bone_marrow.anc_10e9_per_L < 1.5
}
reasons contains "I-6: Hemoglobin < 9 g/dL" if {
    input.patient.bone_marrow
      .hemoglobin_g_per_dL < 9
}
reasons contains "I-9: LVEF < 50%" if {
    input.patient.cardiac.lvef_percent < 50
}
reasons contains "E-1: Autoimmune disease" if {
    input.patient.autoimmune.known_or_suspected
}
reasons contains
  "E-2: Corticosteroids within 28d" if {
    input.patient.medications
      .systemic_corticosteroids_within_28d
}
reasons contains
  "E-6: Brain mets not stable >=1mo" if {
    input.patient.tumor.brain_metastases.present
    input.patient.tumor.brain_metastases.treated
    input.patient.tumor.brain_metastases
      .stable_months < 1
}
```

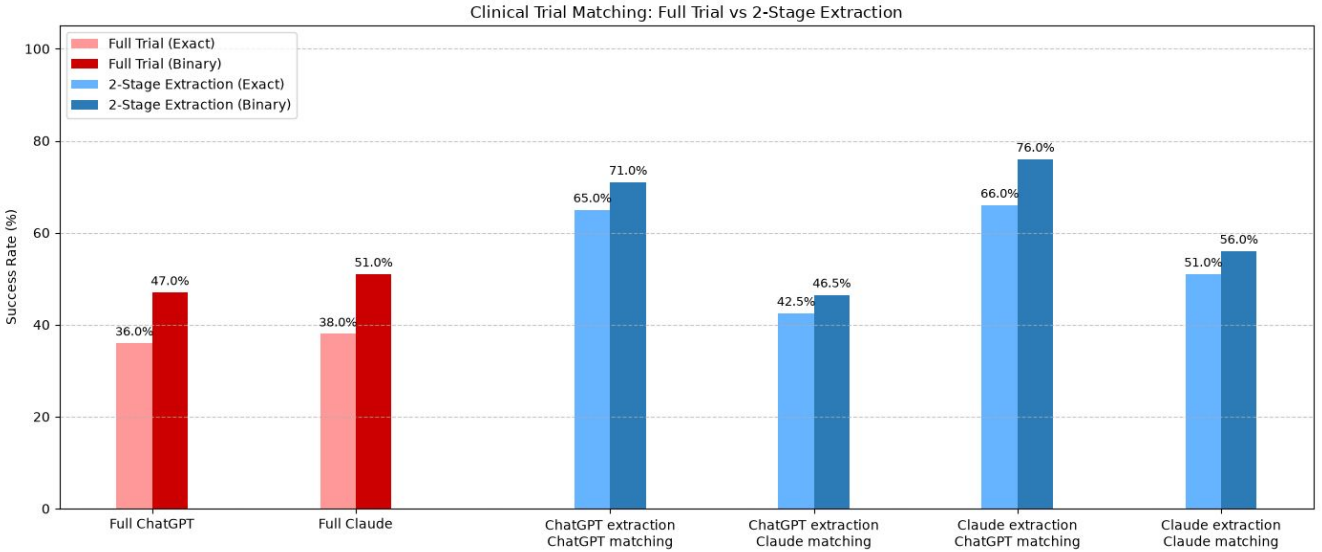
Matchmaking agent

1. Basic matching (Full criteria)
   LLM is provided with the full text of the clinical trial's eligibility criteria and the patient's profile in a single prompt.
   The LLM is asked to make a direct decision.
   Pros: Simple, captures context from the entire criteria block.
   Cons: Large prompts, potential for missing specific constraints in long texts, lack of granularity in explanations.
2. Criteria extraction-based matching
   Step A (Extraction) + Step B (Evaluation)
3. Parameter extraction + rule-based matching

TREC 2021 benchmark

Metric	Mean ± SD	Median	Min	Max
Clinical Trials				
Words	1,917.93 ± 3,993.01	859	120	59,114
XML tags	483.51 ± 1,366.63	112	44	21,941
Distinct XML tags	86.68 ± 29.92	77	41	170
Numeric measurements	208.25 ± 556.81	75.5	21	6,957
Patient Profiles (Topics)				
Words	134.89 ± 36.52	134	51	218
Numeric measurements	9.47 ± 6.35	8	1	27

Accuracy on TREC dataset



Threat to validity / limitation:

Trials are real but patient profiles are generated in TREC dataset

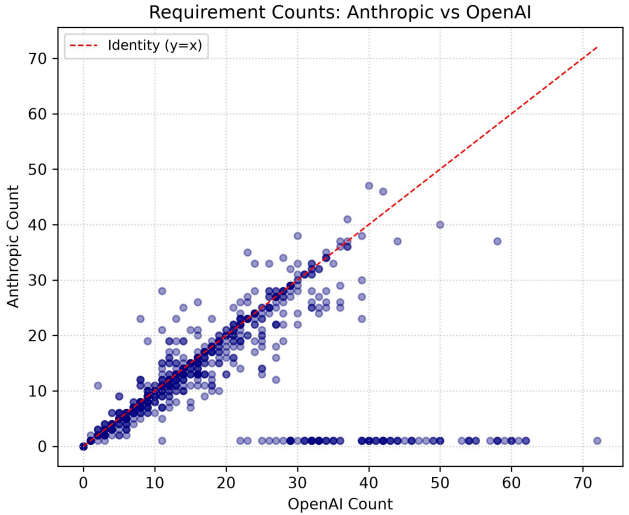
Data for decision making is missing in profiles

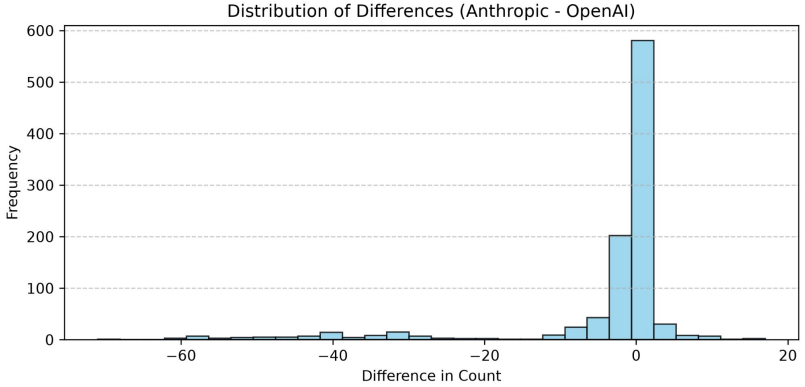
“ground truth” may not be 100% accurate

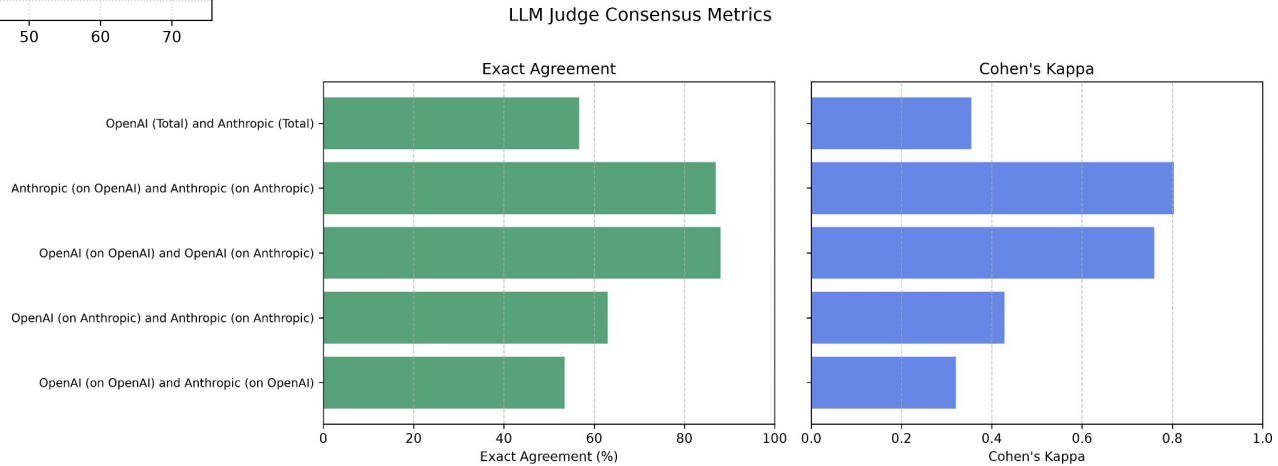
Value extraction (Precidio dataset)

Model	Accuracy
OpenAI GPT-5	94.36
Anthropic Claude 4.6	90.77

Consensus


Requirement Counts: Anthropic vs OpenAI
Identity (y=x)
Anthropic Count
OpenAI Count
0
10
20
30
40
50
60
70


Distribution of Differences (Anthropic - OpenAI)
Frequency
Difference in Count
0
100
200
300
400
500
600
−60
−40
−20
0
20


LLM Judge Consensus Metrics
Exact Agreement
Cohen's Kappa
OpenAI (Total) and Anthropic (Total)
Anthropic (on OpenAI) and Anthropic (on Anthropic)
OpenAI (on OpenAI) and OpenAI (on Anthropic)
OpenAI (on Anthropic) and Anthropic (on Anthropic)
OpenAI (on OpenAI) and Anthropic (on OpenAI)
Exact Agreement (%)
Cohen's Kappa
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0

No/low-code Data-Agnostic ML & AI

Workflows across heterogeneous data sources and formats

- **Data-agnostic:** decouple AI/ML reasoning from specific datasets, schemas, domains, or providers.
- **Heterogeneous inputs:** support diverse formats, schemas, vocabularies, modalities, and completeness.
- **Reusable methods:** apply the same reasoning and analysis components across datasets and use cases.
- **Robust & generalisable:** handle missing, inconsistent, uncertain, and evolving data.
- **Composable workflows:** agents dynamically select and orchestrate capabilities based on available data.

Agentic Decision Support

- Agents that can make decisions across changing data environments
- **Key challenge:** ensuring that *data independence does not become decision unreliability* → agents need explicit schemas, provenance, validation, uncertainty handling and verifiable decision mechanisms.
- Federated outcome prediction for patients (multi-modal and explainable)

Future work

Agentic decision support: Federated **causal treatment-effect estimation**

Statistical models estimate treatment outcomes and **uncertainty** across institutions.

Formal rules determine which treatments are clinically admissible and sufficiently supported.

Three questions:

1. formalising clinical guidelines
2. incorporating multiple outcomes and patient preferences
3. handling uncertainty and cross-site transportability

The system can abstain when evidence is insufficient and should provide an auditable set of supported interventions rather than relying on an opaque AI recommendation.