

A Cross-Platform Benchmark Framework for Mobile Semantic Web Reasoning Engines

William Van Woensel, Newres Al Haider, Ahmad Marwan Ahmad,
Syed Sibte Raza Abidi

Introduction (1)



- Semantic Web technology
 - Used extensively in **healthcare domain**
 - Facilitates data integration
 - SNOMED, BioPortal, ..
 - Allows expressive, rule-based reasoning
 - SWRL, SPIN, RuleML
- Clinical Decision Support Systems (CDSS)
 - Apply rule-based reasoning to infer clinical data
 - Application logic <> Domain knowledge

Introduction (2)



- Mobile deployment of CDSS
 - Enabled by improvements in mobile hardware
 - Improve timeliness of alerts
 - Range of health data is collected locally
 - E.g., BP, HR (also falls, stroke, ..)
 - *Local DSS*: issue alerts, even when lacking connectivity
 - Avoid privacy issues
 - *Local DSS*: avoid sending privacy-sensitive data
- .. requires mobile deployment of SemWeb reasoning

*E.g., Apple HealthKit,
Google Fit, ..*

Challenges



- Limited mobile computing resources
 - ➔ Enable benchmarking of current systems
 - Use existing, real-world data- & rule-set (varying scale & complexity)
 - Use existing reasoning process flows
- Heterogeneity of mobile platforms
 - Android, iOS, BlackBerry, Windows Phone, Symbian, ..
 - ➔ Enable cross-platform benchmark deployment



Cross-platform Benchmark Framework

1) Generic, standards-based Semantic Web layer

- *Uniform interface*: SPARQL Inferencing Notation (SPIN) & RDF
- Re-use existing, real-world data- & rule-set

2) Benchmark Engine

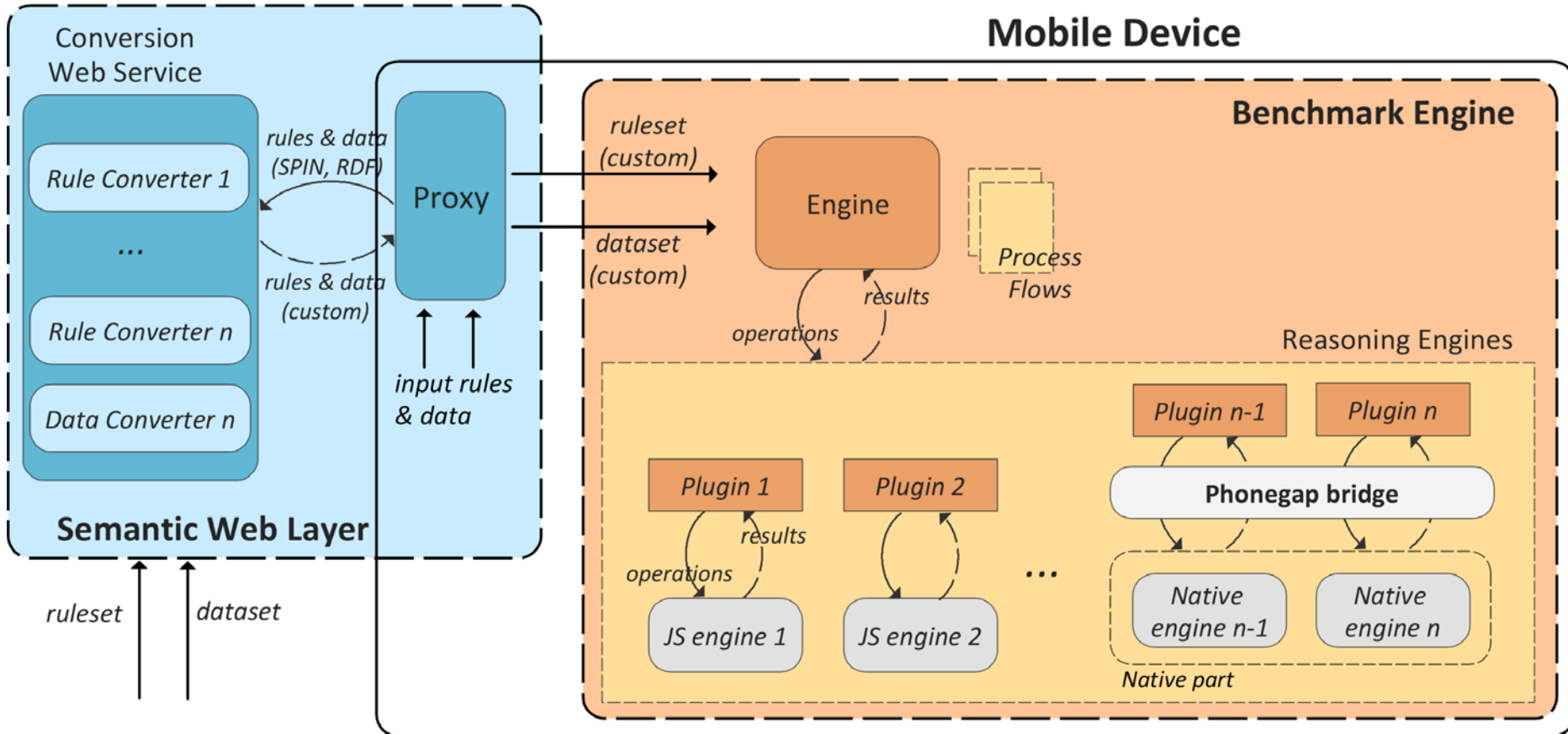
- Investigate & compare mobile rule-based reasoners
- Use existing reasoning process flows

Extensible: plugin new rule/data formats, reasoning engines

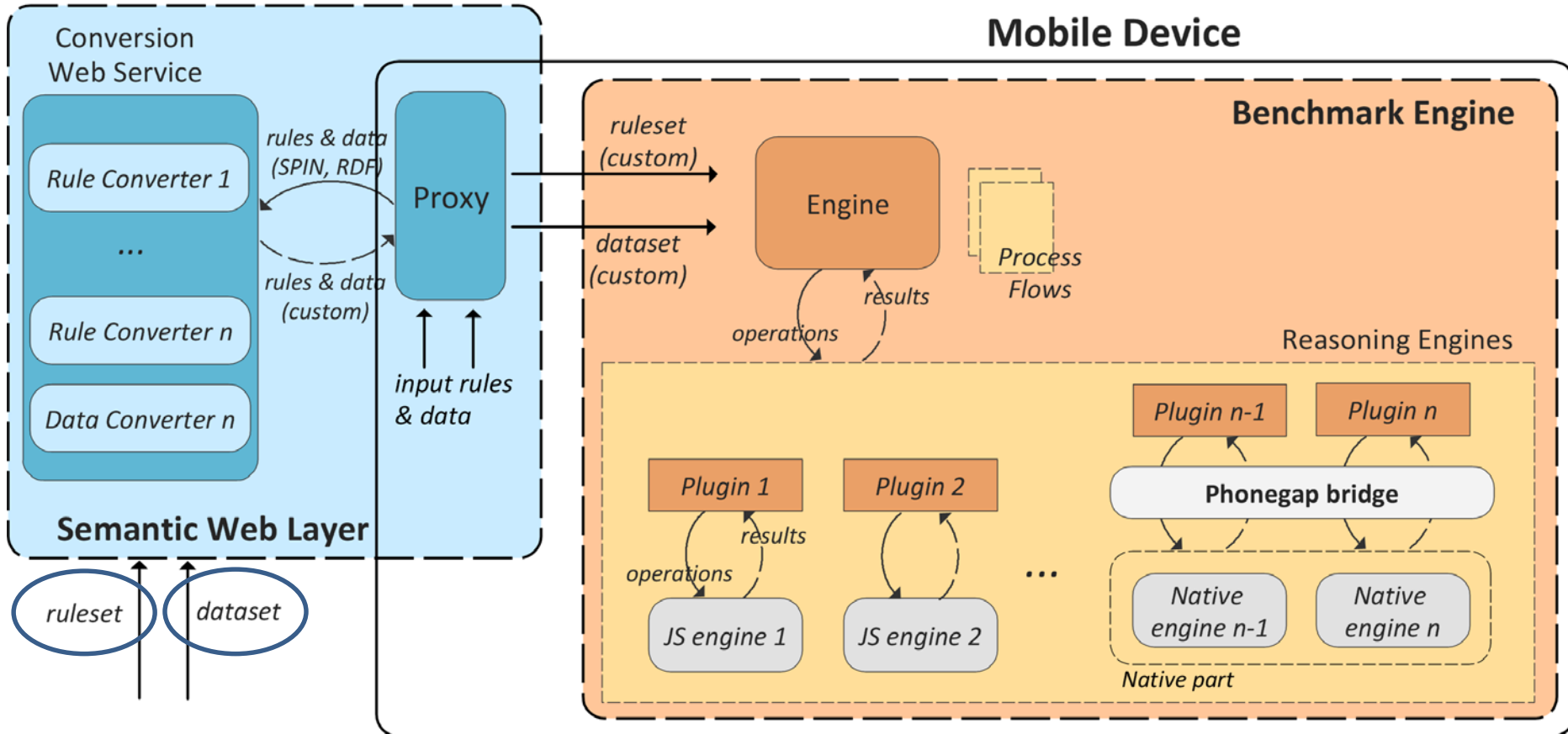
Cross-platform: uses PhoneGap development tool

- Allows deploying mobile JavaScript apps as native apps (Android, iOS, ..)
- Benchmark both JavaScript and native rule systems

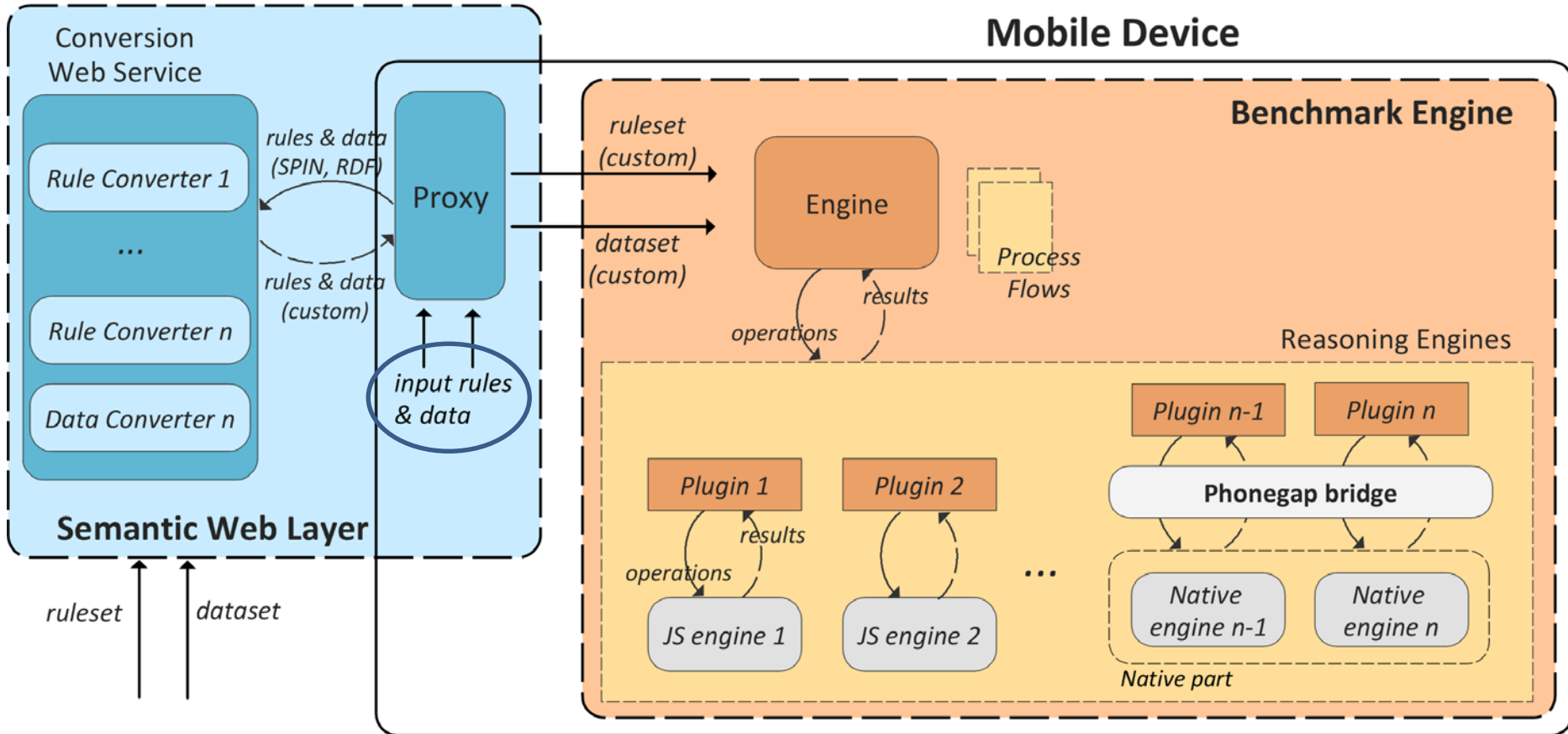
Architecture



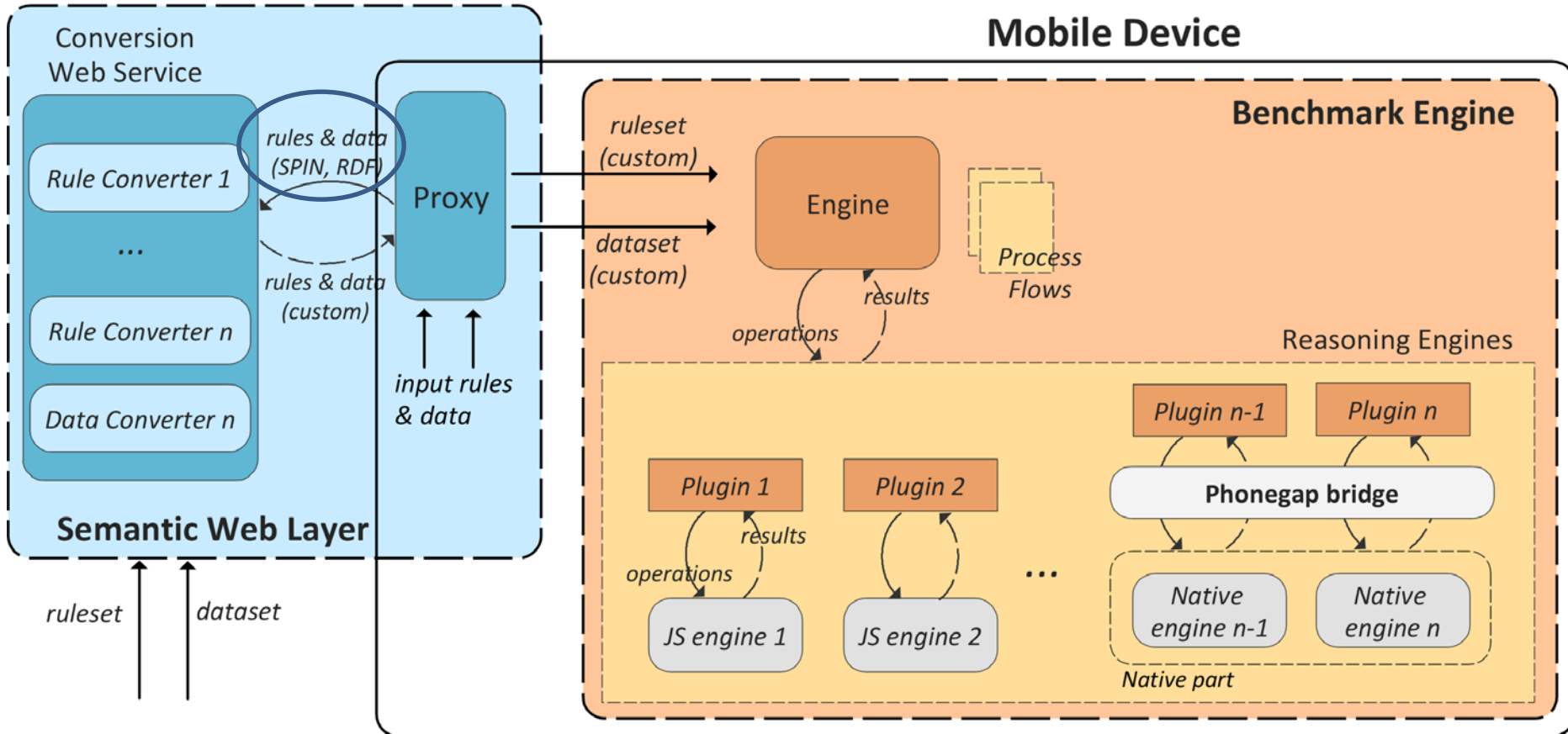
Architecture



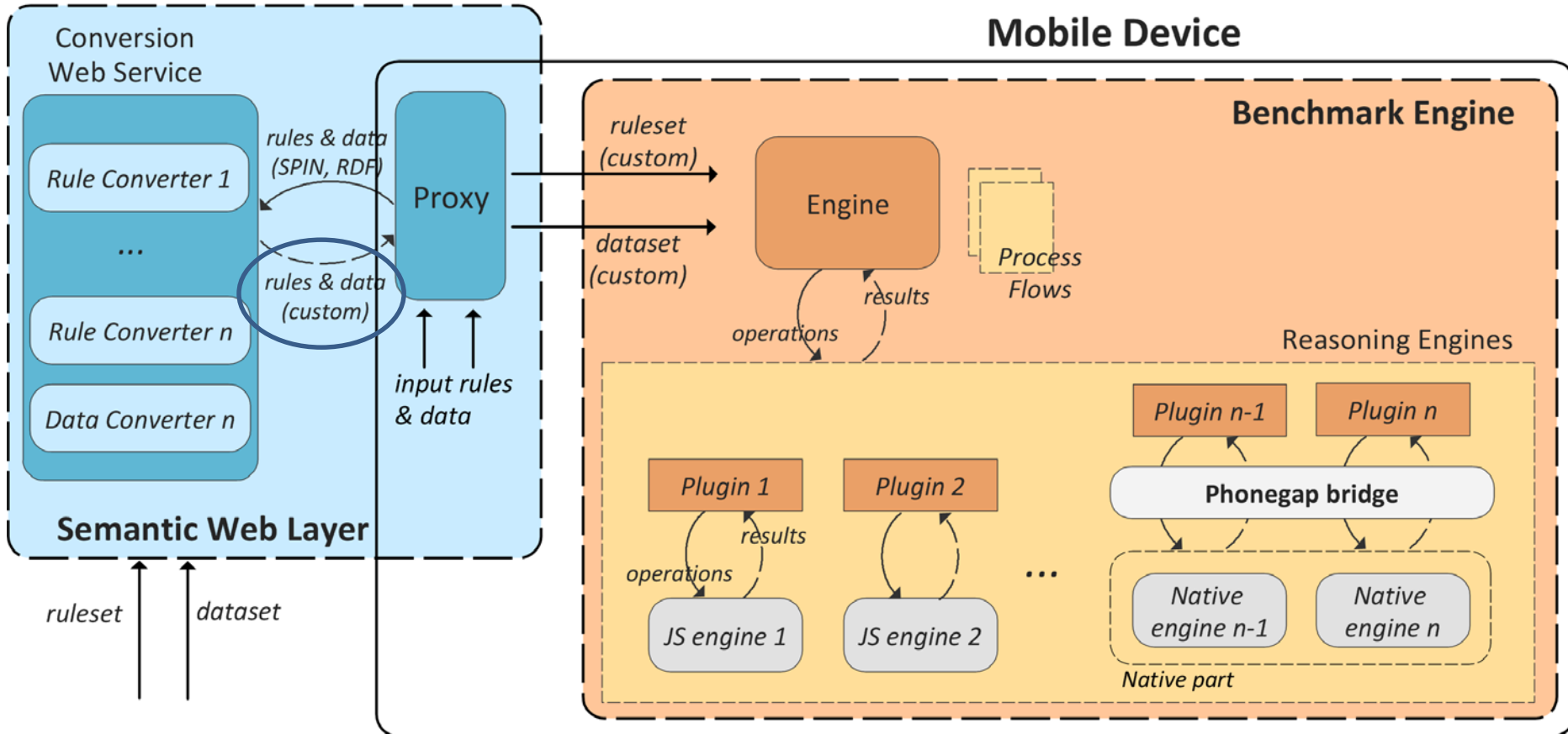
Architecture



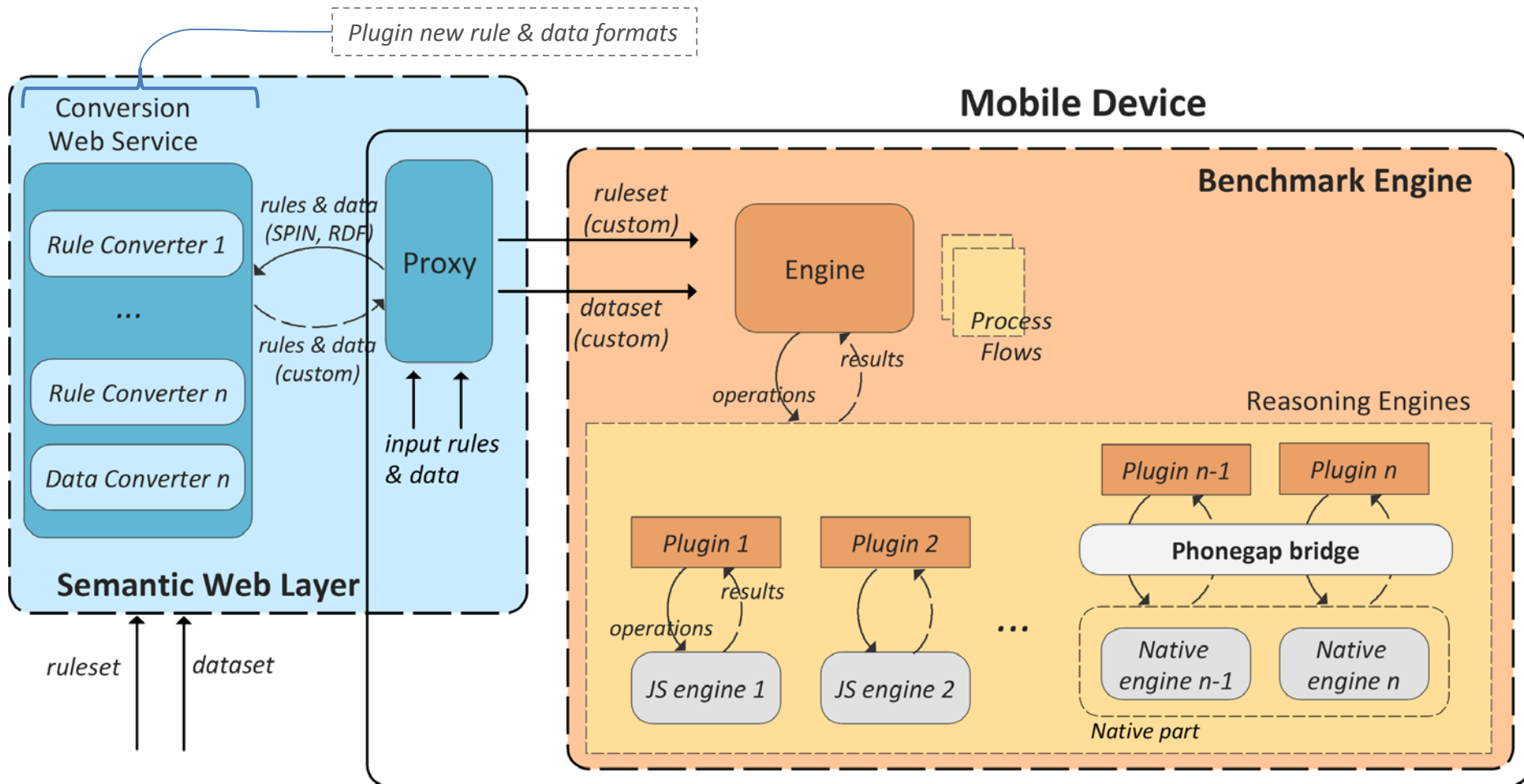
Architecture



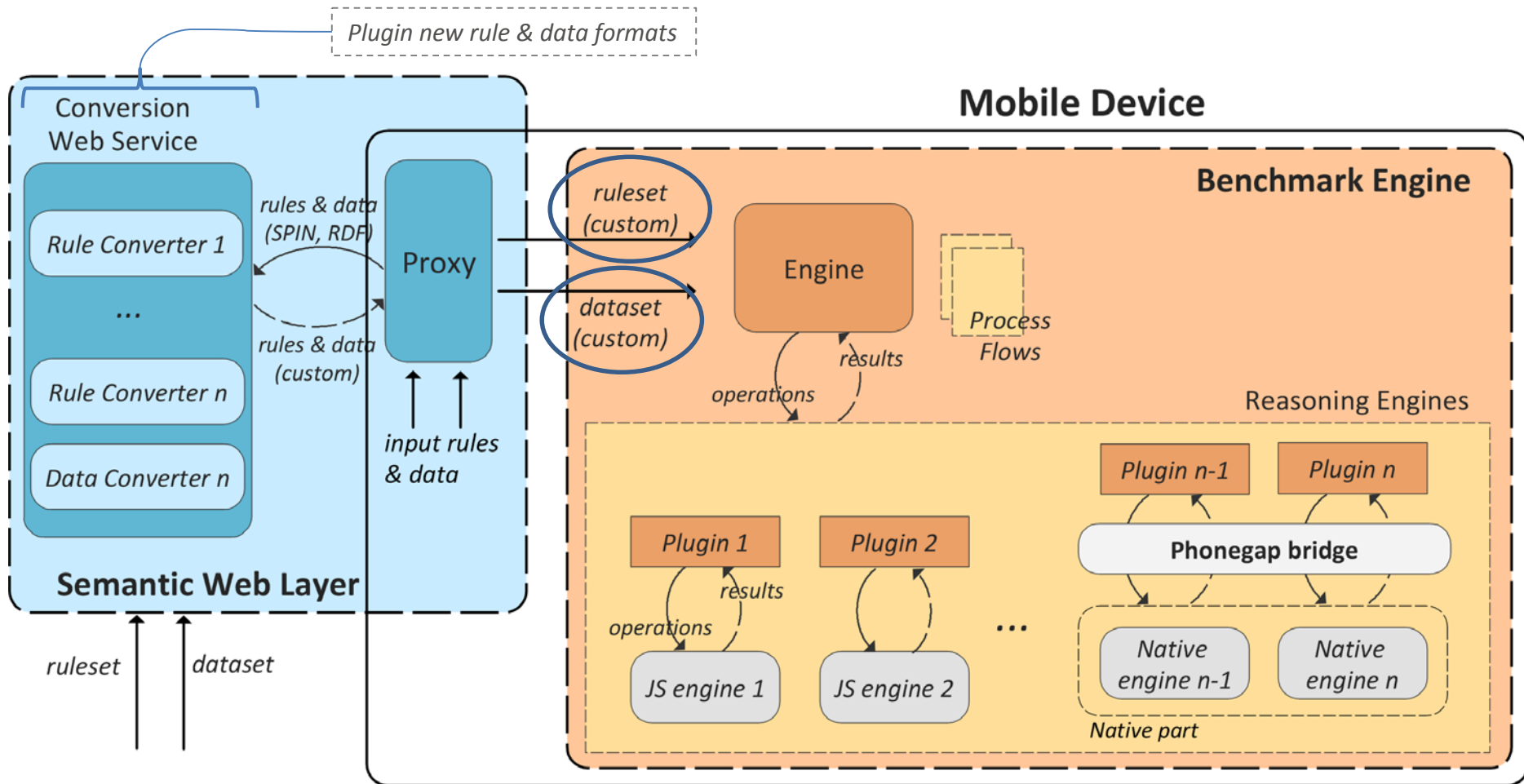
Architecture



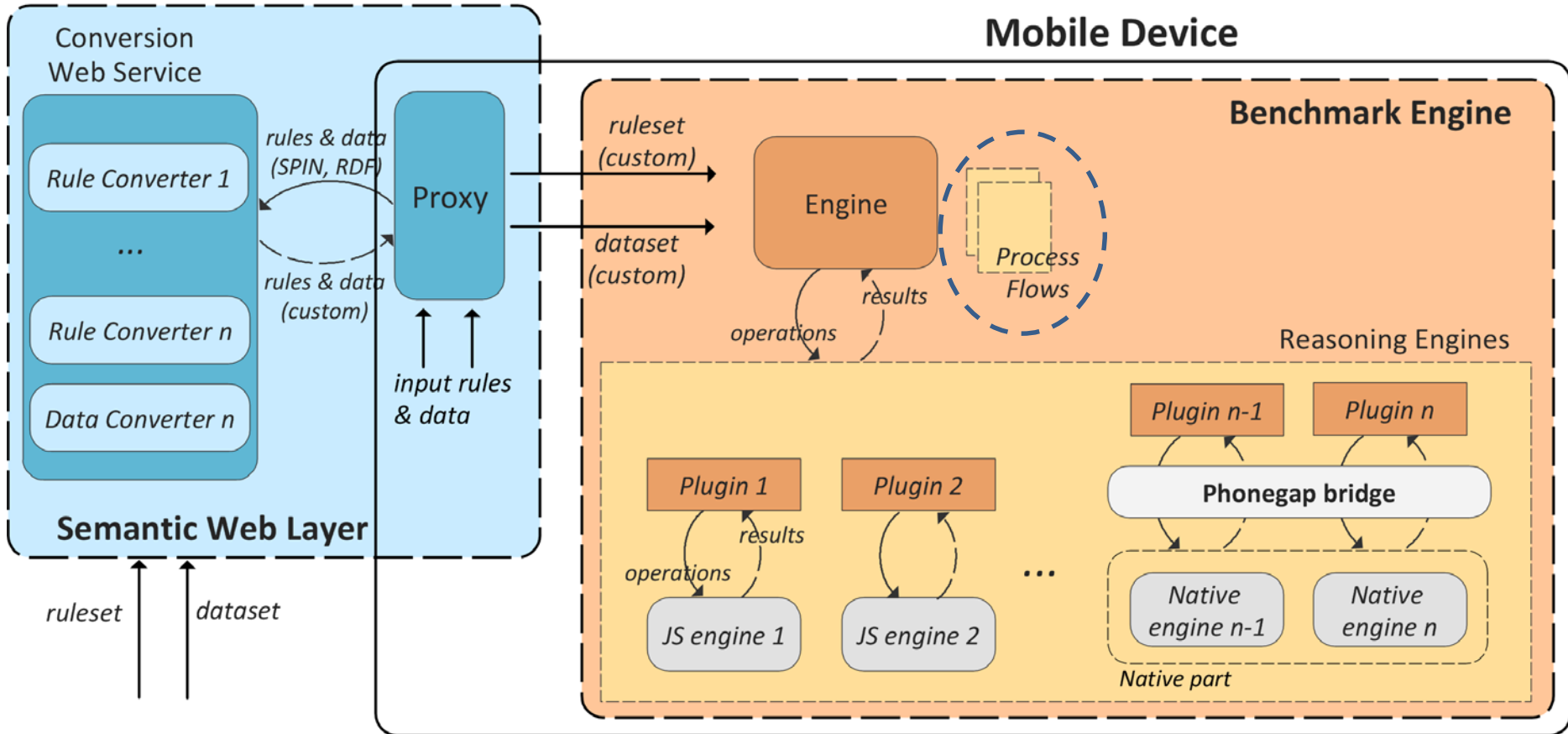
Architecture



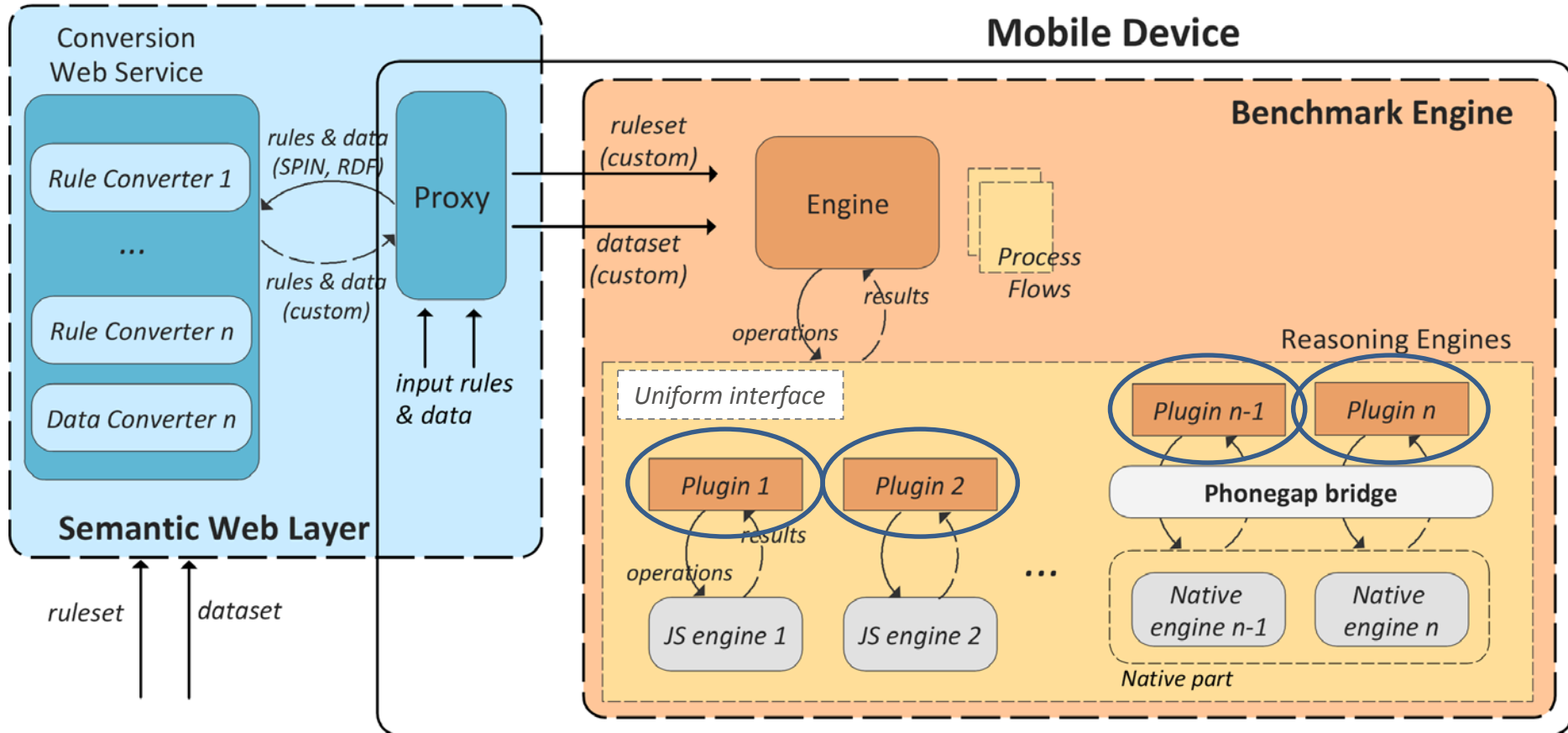
Architecture



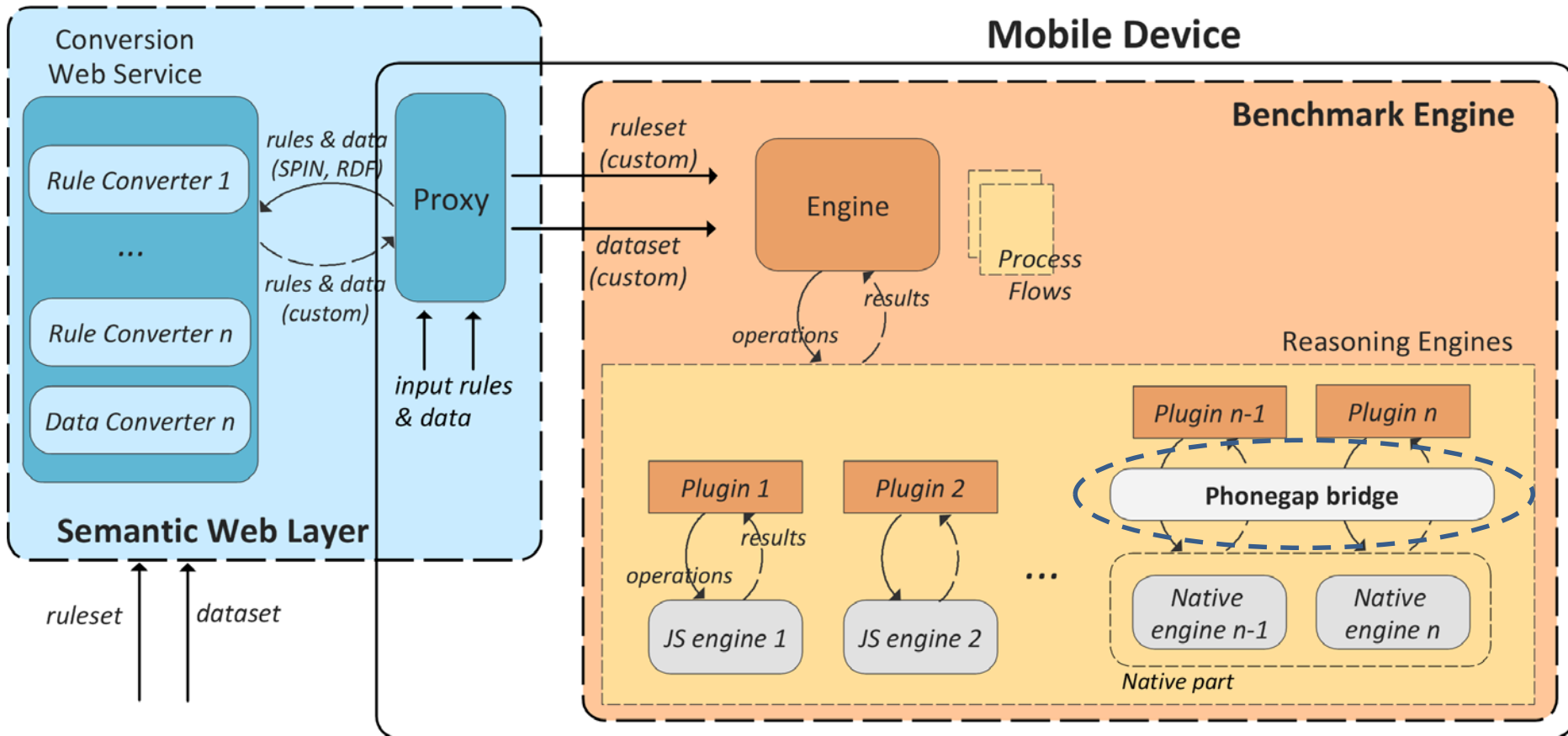
Architecture



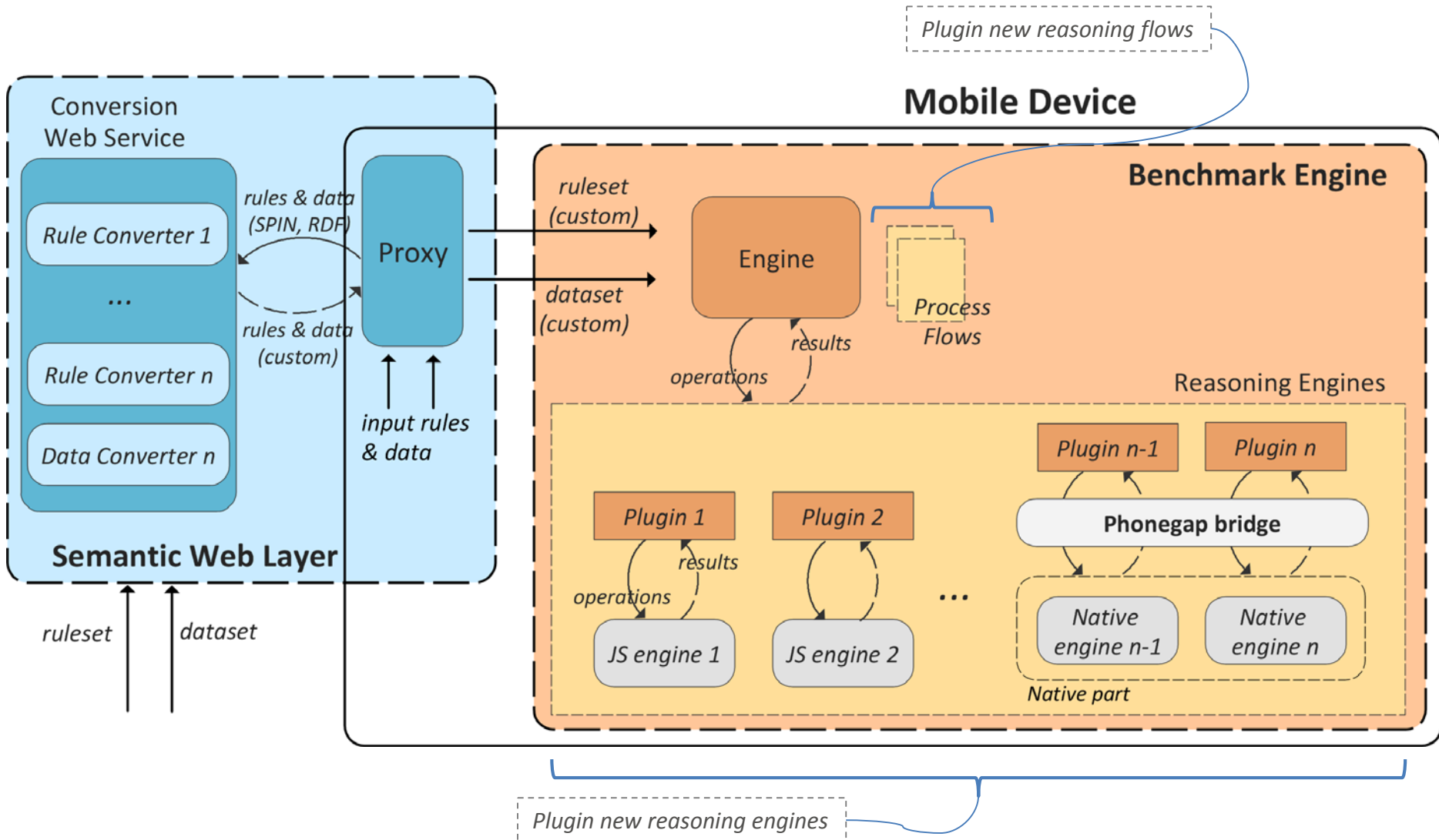
Architecture



Architecture



Architecture



Reasoning Process Flows



- Frequent Reasoning
 - Continuously store collected facts
 - Apply reasoning after timespan X
 - *Re-load engine in memory each time*
- Incremental Reasoning
 - Load store with baseline data
 - Including profile, historical data, ..
 - Apply reasoning after new fact
 - And, on start-up
 - *Keep engine continuously in-memory*

Supported Systems



- AndroJena
 - RDFQuery
 - RDFStore-JS
 - Extended with (naïve) reasoning
 - Nools
 - Extended with Semantic Web layer
- Native
(Android)
- JavaScript

Conclusion & Future work



Conclusion

- Mobile benchmark framework for mobile, Semantic Web reasoning engines
 - Easily extensible, cross-platform

Future work

- Apply optimization techniques for rulesets
 - E.g., based on RETE or from SPARQL query optimization
 - Tests already show large impact on performance

Last Slide



- Benchmark Framework

https://niche.cs.dal.ca/benchmark_framework/

(underscore)

- **Contact:** William Van Woensel

william.van.woensel@dal.ca

Proof-of-concept Benchmark



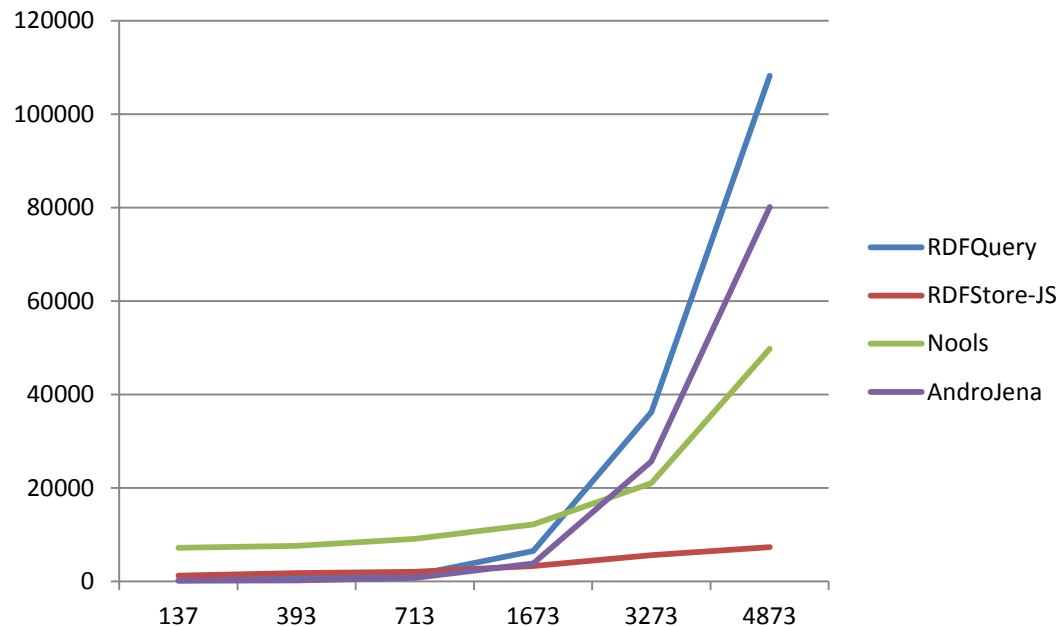
- **Ruleset:** derived from AF treatment guidelines
 - Total of 10 rules
- **Dataset:** comprises clinical info
 - Personal clinical information
 - Health measurements & observations (facts)
 - Increasing dataset sizes to test scalability
 - 1 - 75 facts; 137 – 4873 triples

Benchmark: Frequent Reasoning



➤ Loading entire dataset; execute all rules

• Total times:



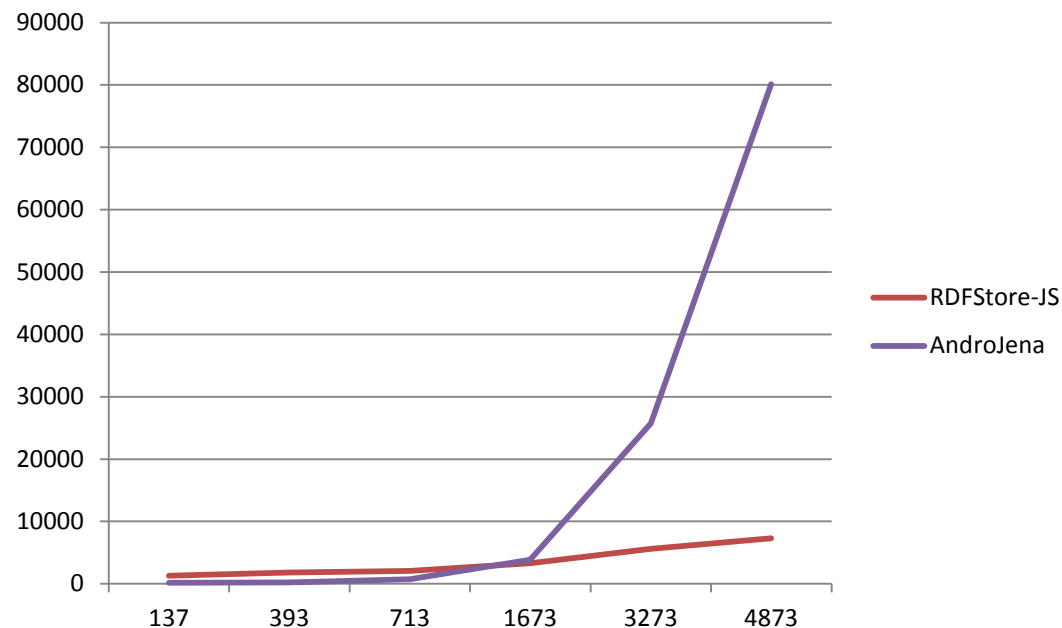
- Overall, **RDFStore-JS** is most scalable solution
 - Performs better for larger datasets (> 1673 triples)
- **AndroJena**: better for smaller datasets

Benchmark: Frequent Reasoning



➤ Loading entire dataset; execute all rules

• Total times:



- Overall, **RDFStore-JS** is most scalable solution
 - Performs better for larger datasets (> 1673 triples)
- **AndroJena**: better for smaller datasets

Benchmark: Incremental Reasoning



- Load baseline dataset (25 facts; 1673 triples)
- Load single fact (5 triples) ; execute all rules

- Total times:

	RDFQuery	RDFStore-JS	Nools	AndroJena
loading	38	8	15	18
reasoning	6097	1440	12	3403
total	6135	1448	27	3421

- **Nools**: best option for small rule & datasets
 - Low reasoning times makes up for initialization times
- **RDFStore-JS**: still most scalable option

Nools performs extremely well
Others: times are comparable to Frequent Reasoning