



Architecture Review

Fatafat

February 12 2015
Version 1.0
ligaDATA



ligaDATA

Agenda

- Introducing Fatafat
 - A Simple Use-Case
 - Architecture
 - Metadata API
 - Model workflow (DAG)
 - Performance & Scale
-

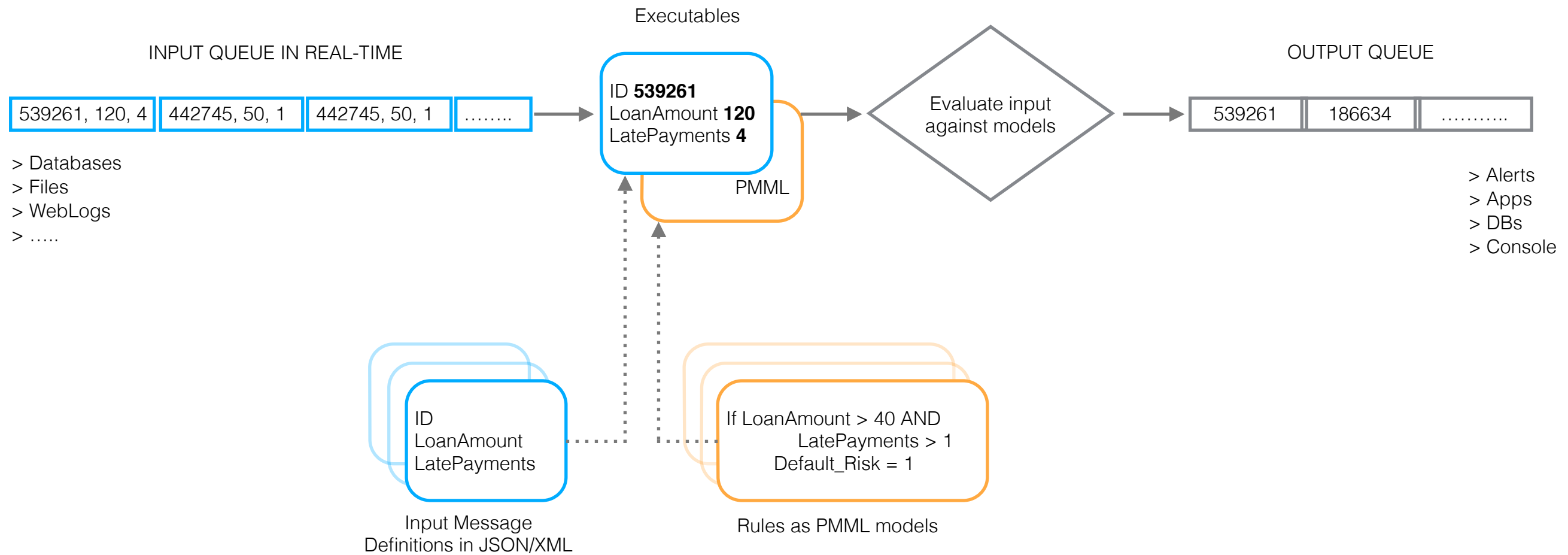
Introducing Fatafat

Fatafat

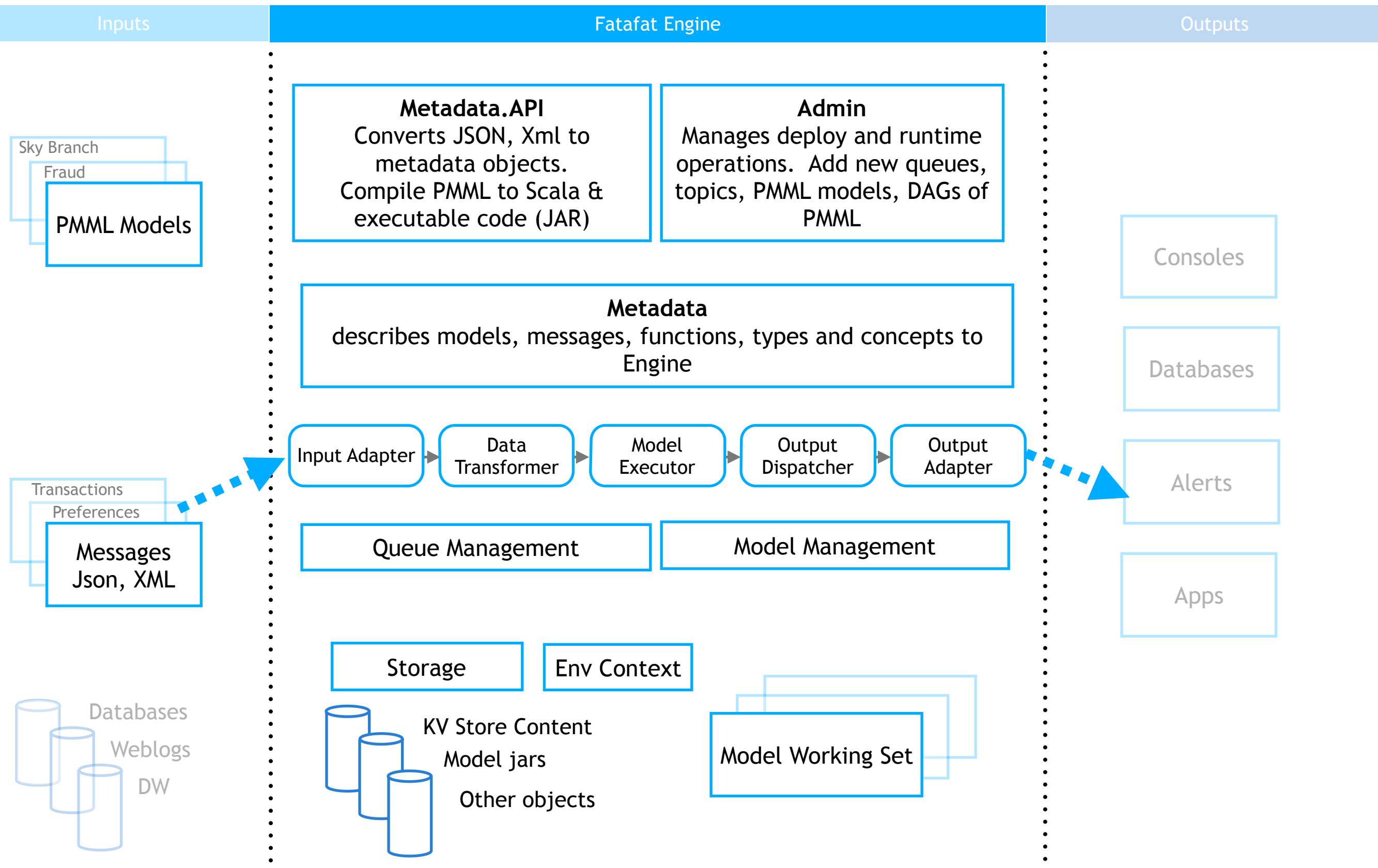
- Is an open-source decision engine that provides real-time processing of big data streams
 - Helps act upon opportunities or threats based on data science, business rules and historical data
 - Provides unmatched efficiency and scale via cluster based deployment
 - Guarantees exactly once execution of each input message
 - Reduces PMML model deployment from ~months to ~days
 - Is easy to deploy and comes with samples, guides, and a vibrant community right out of the box
 - Is based on open-source tools and technology stack
-

A Simple Use-Case

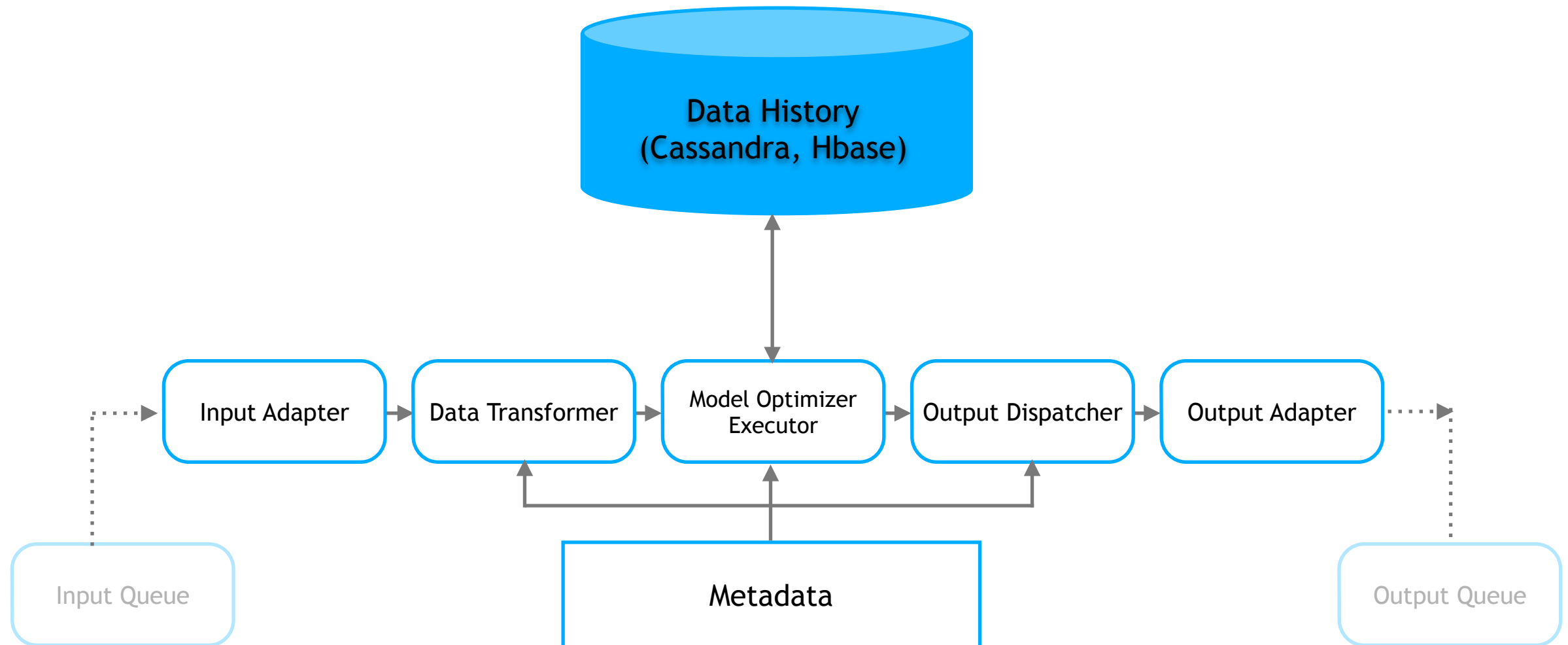
Who is at a risk of defaulting based on loan amount and late payments?



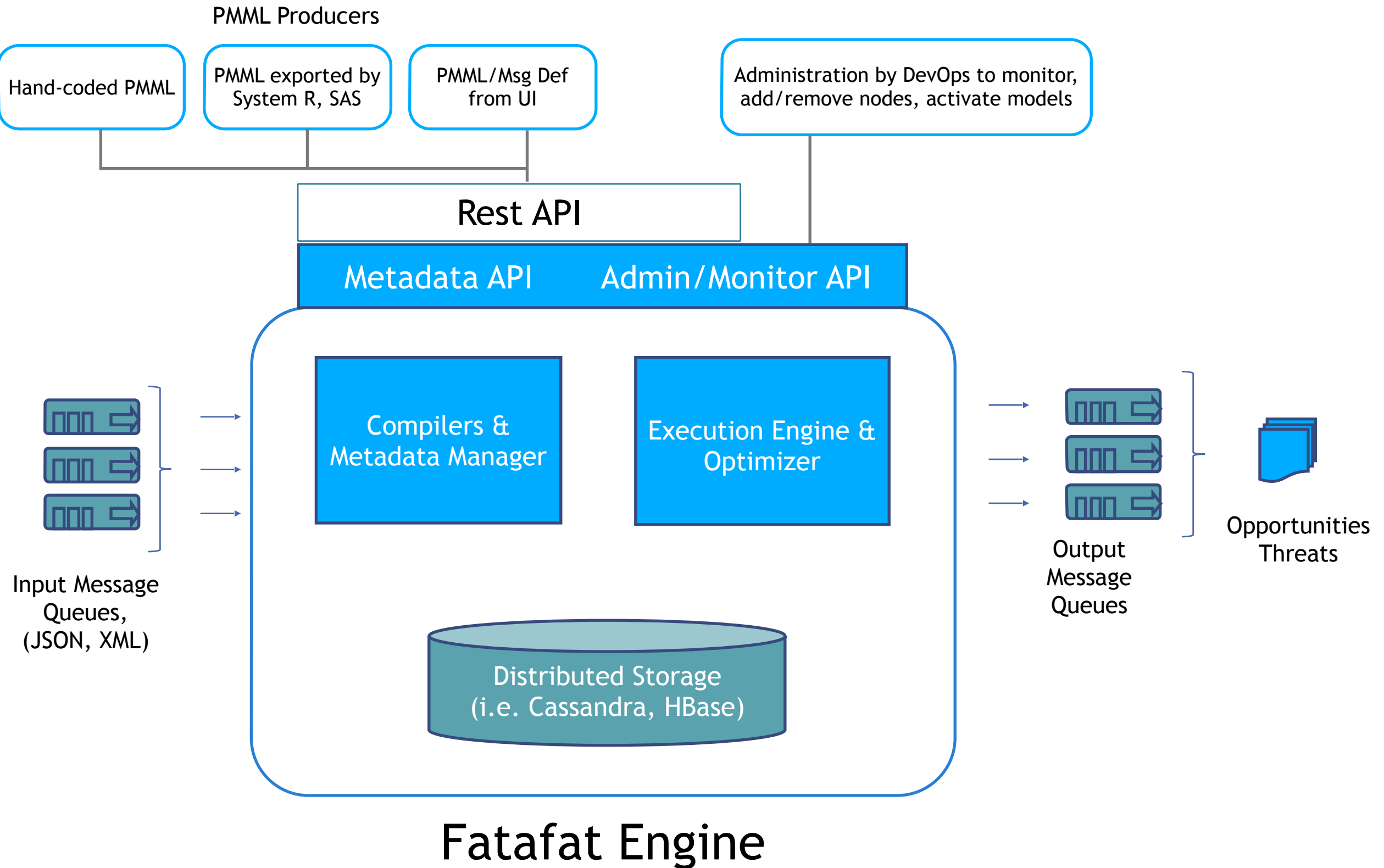
Architecture



Engine Execution Flow



Metadata API Sub-Systems



Metadata API Elements

Models

(PMML Rule Set, i.e.
fraud, attrition)

Functions

(in PMML, Scala, user
defined)

Messages

(from input queue, i.e.
Kafka or MQ)

Containers

(i.e. a record or
dimensions)

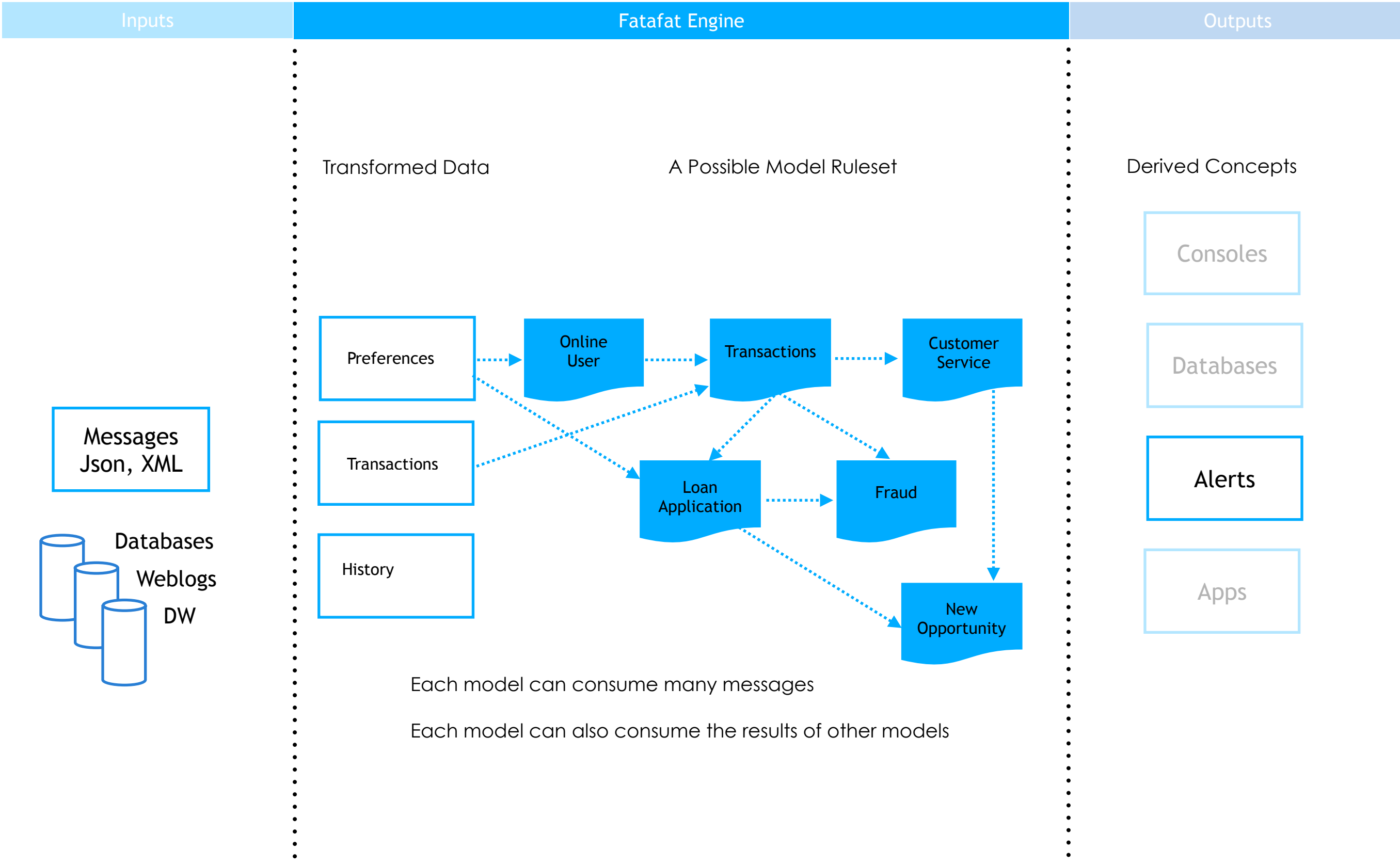
Types

(i.e. array of customers,
loan applicants, types of
containers)

Concepts

(PMML Output fields,
preprocessing, scores)

Model Workflow



Performance & Scale

Fatafat on a **single** node: **220k to 230k messages / second**

CONFIGURATION:

16 core box, using Solid State Disc

Sample Tool to generate messages of size 1k (not being reduced)

Data Mining uses 100's to 100k fields – not 100 byte message

Kafka Queue - 3 input queues, each queue with 8 partitions

Fatafat Engine - Using the remaining 12-13 cores

COMPARISON:

Storm is currently the lowest latency Apache big data system

Storm integration, got up to 90k to 100k for same data

Fatafat is 2.4 times faster than Storm = (225k/95k) in this test

SCALE:

Fatafat can horizontally scale while keeping the same performance.

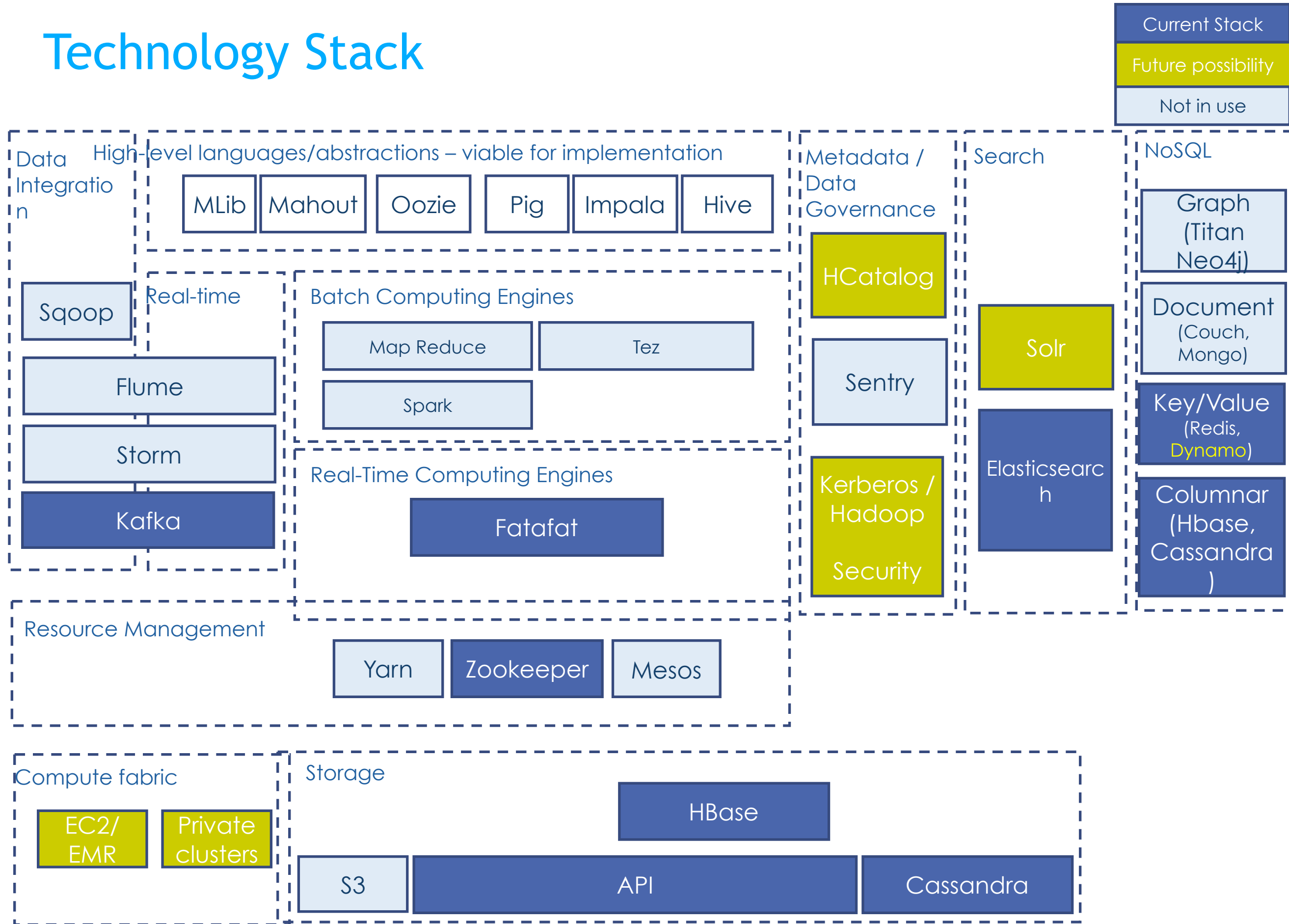
Performance & Scale

Fatafat on a **single** node: **220k to 230k messages / second**

How do we do it fast:

- Message partitioning
 - Dimension caching
 - DAG execution on a single node
 - Efficient communication between dependent models
 - Parallel execution
 - Multi-node optimization
-

Technology Stack



Questions