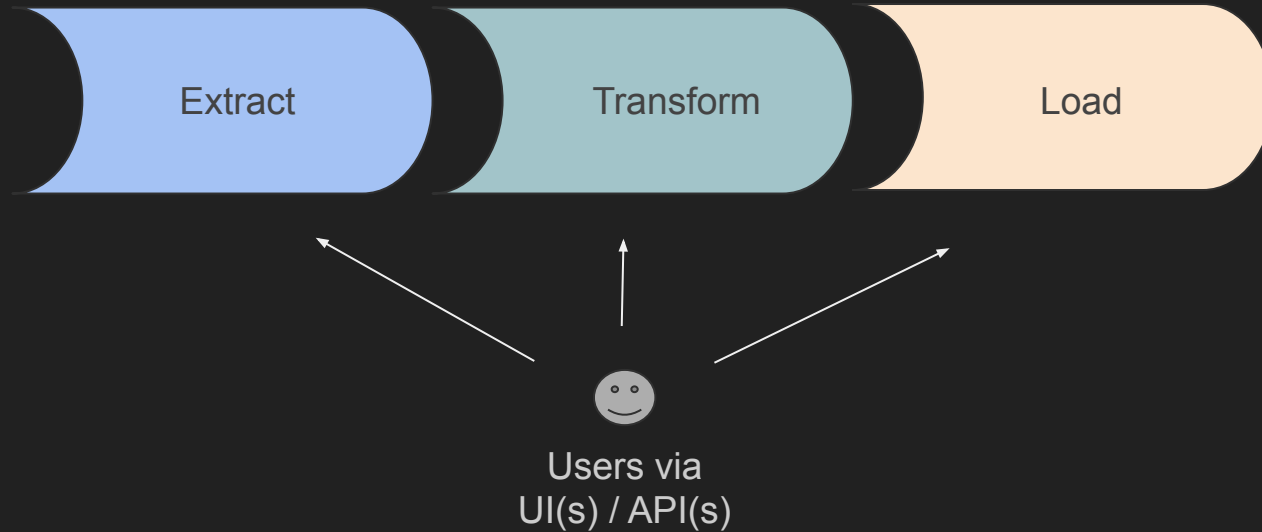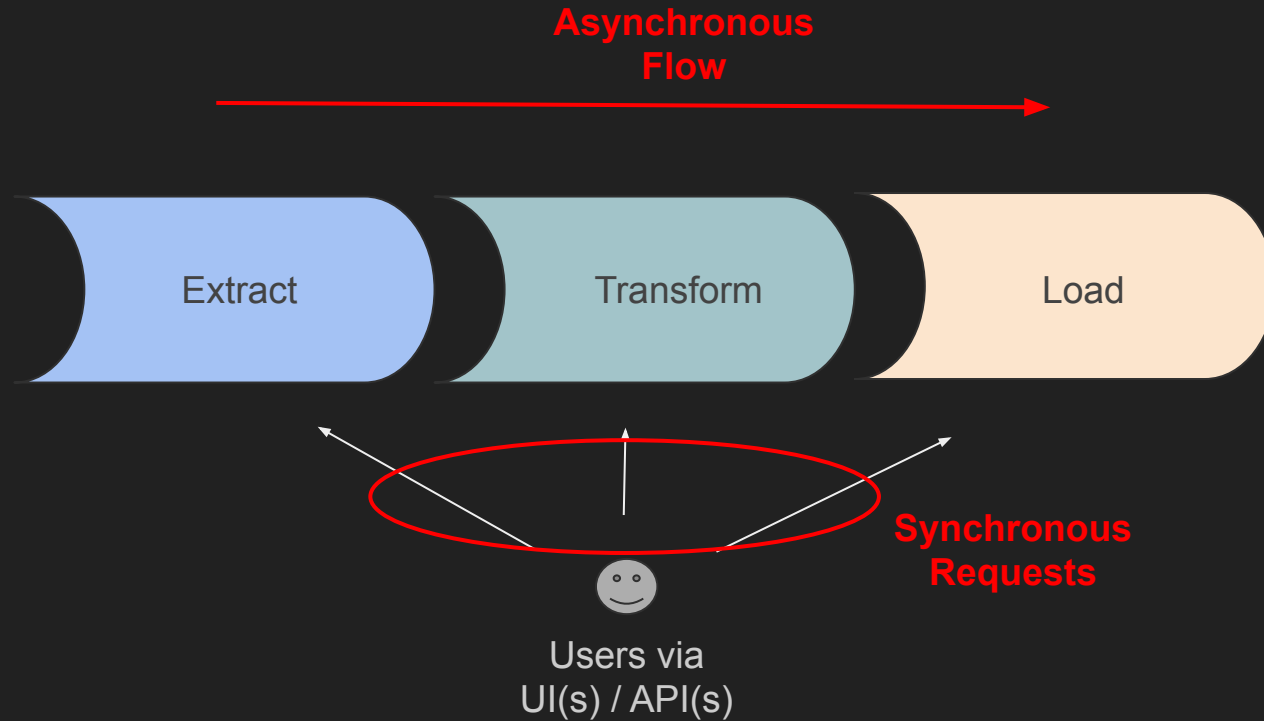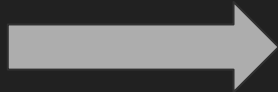# ODS Architecture

An Overview

System Metaphor

System Metaphor
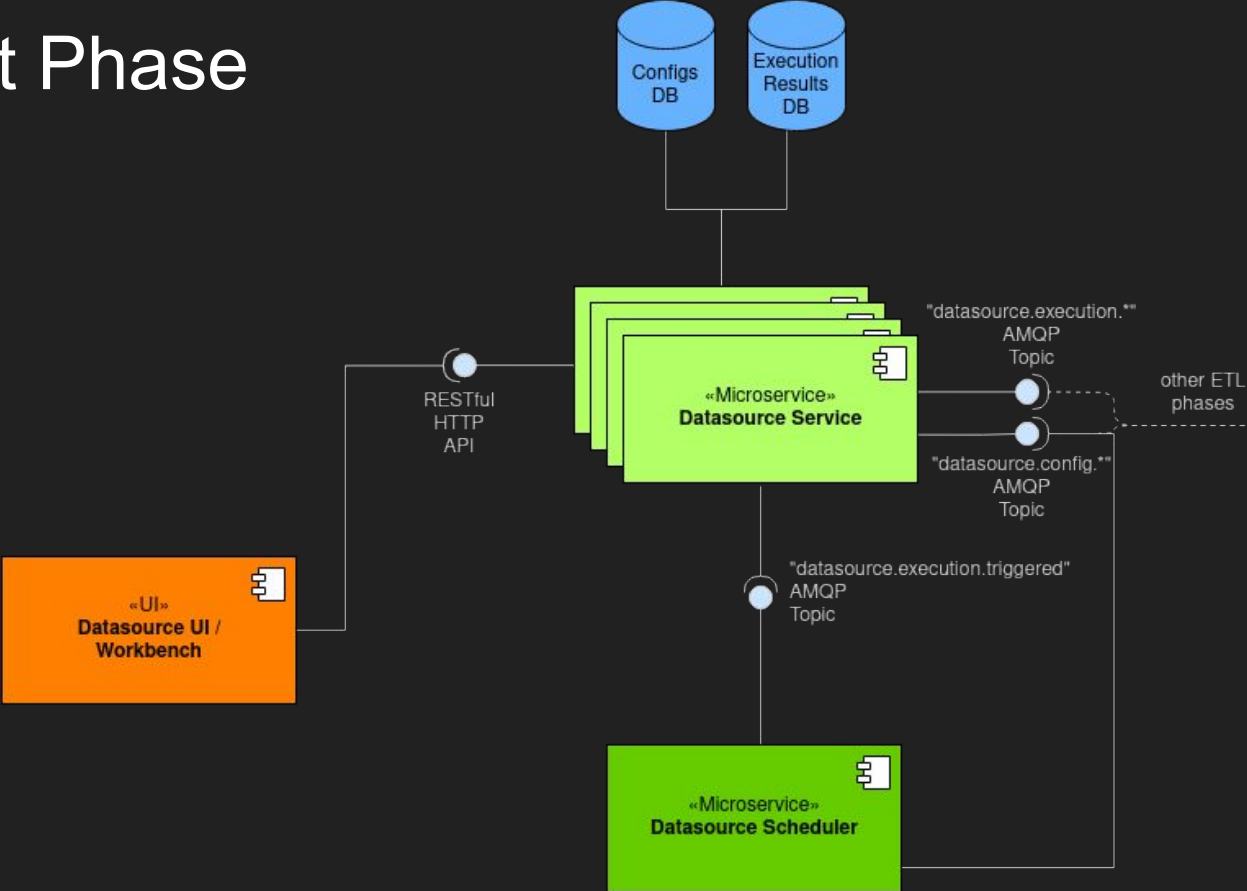
Architecture Requirements

- Scale ETL phases individually
- Failing ETL jobs are not an exception => don't propagate faults
- Unpredictable tech stack (esp. sandboxing for transform phase)
- Multiple repositories with different licenses due to open source strategy
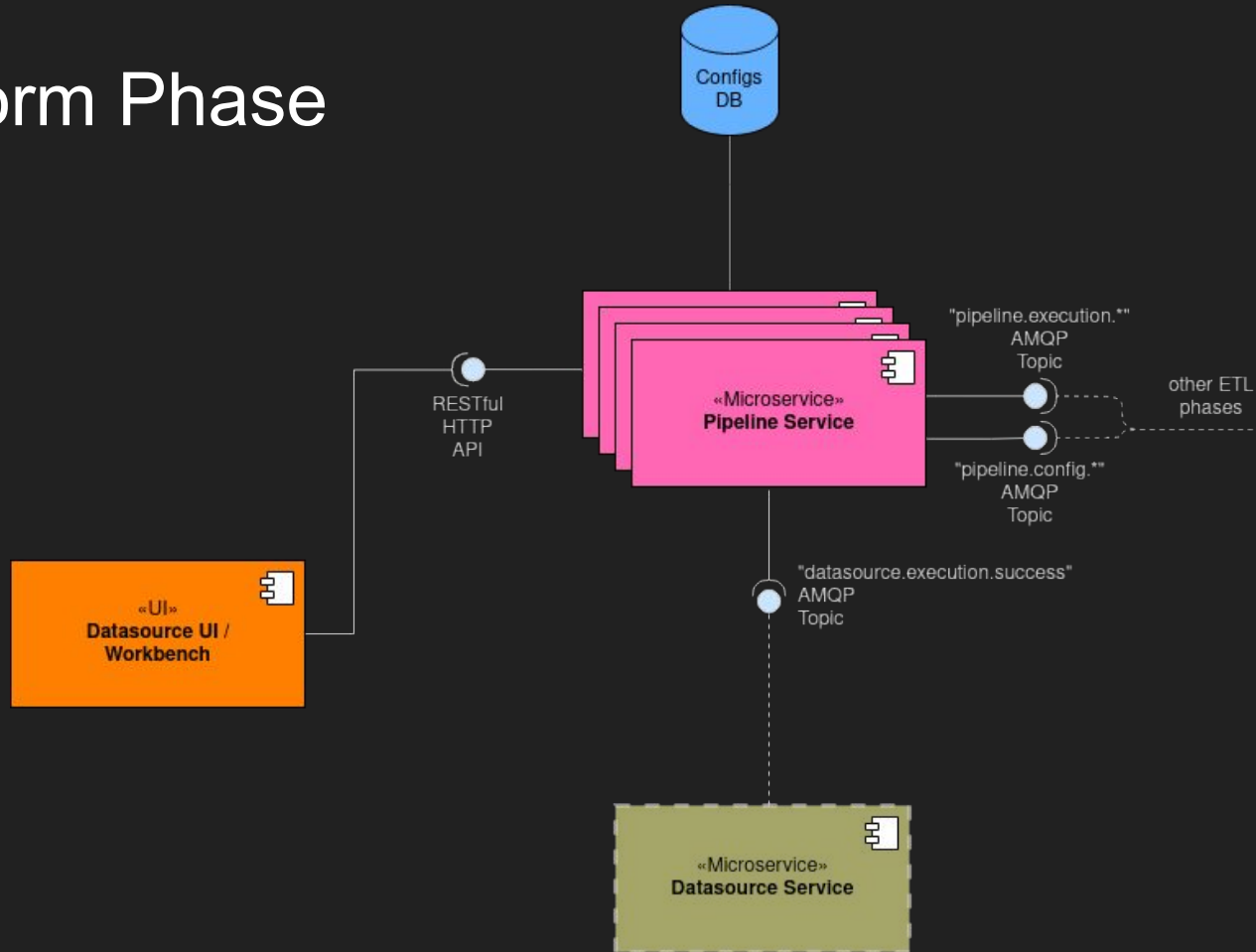- Eventual consistency is ok since ETL phases are asynchronous

Microservices (light)

- Not yet focus on independent deployability
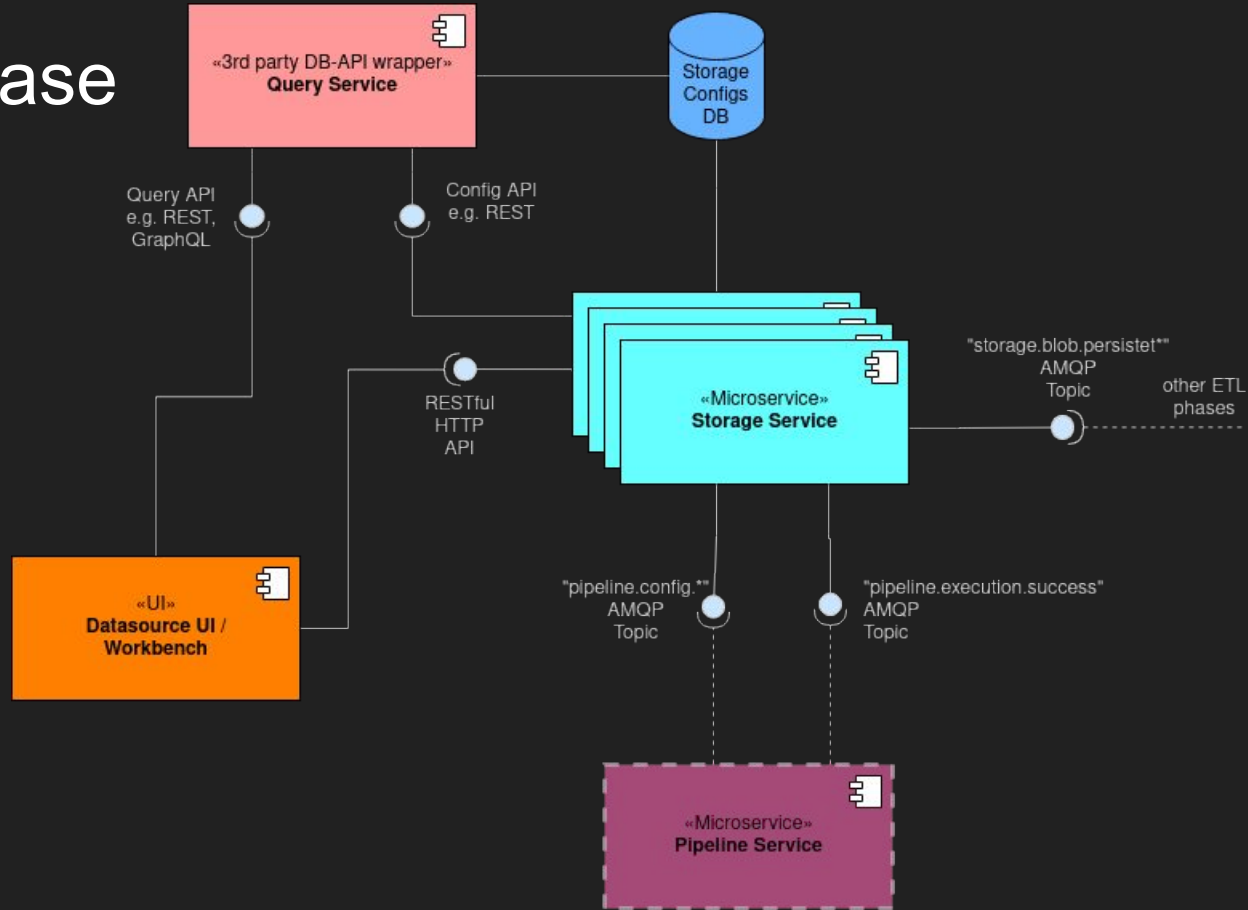- Not yet battle-tested towards scalability

Extract Phase

Transform Phase

Load Phase

Notification Phase

- Downstream notification mechanism
  - Webhooks
  - Slack
  - Google Firebase Notifications (=> mobile push notifications)

- Architecture / API similar to other services
  - Synchronous configuration via RESTful API
  - Asynchronously triggered by domain-based AMQP events

Questions?

https://github.com/jvalue/open-data-service