MODESTI

Monitoring Data Entry System for Technical Infrastructure

Justin Lewis Salmon
BE-ICS-SSE

Outline

- Purpose of MODESTI

- Architecture

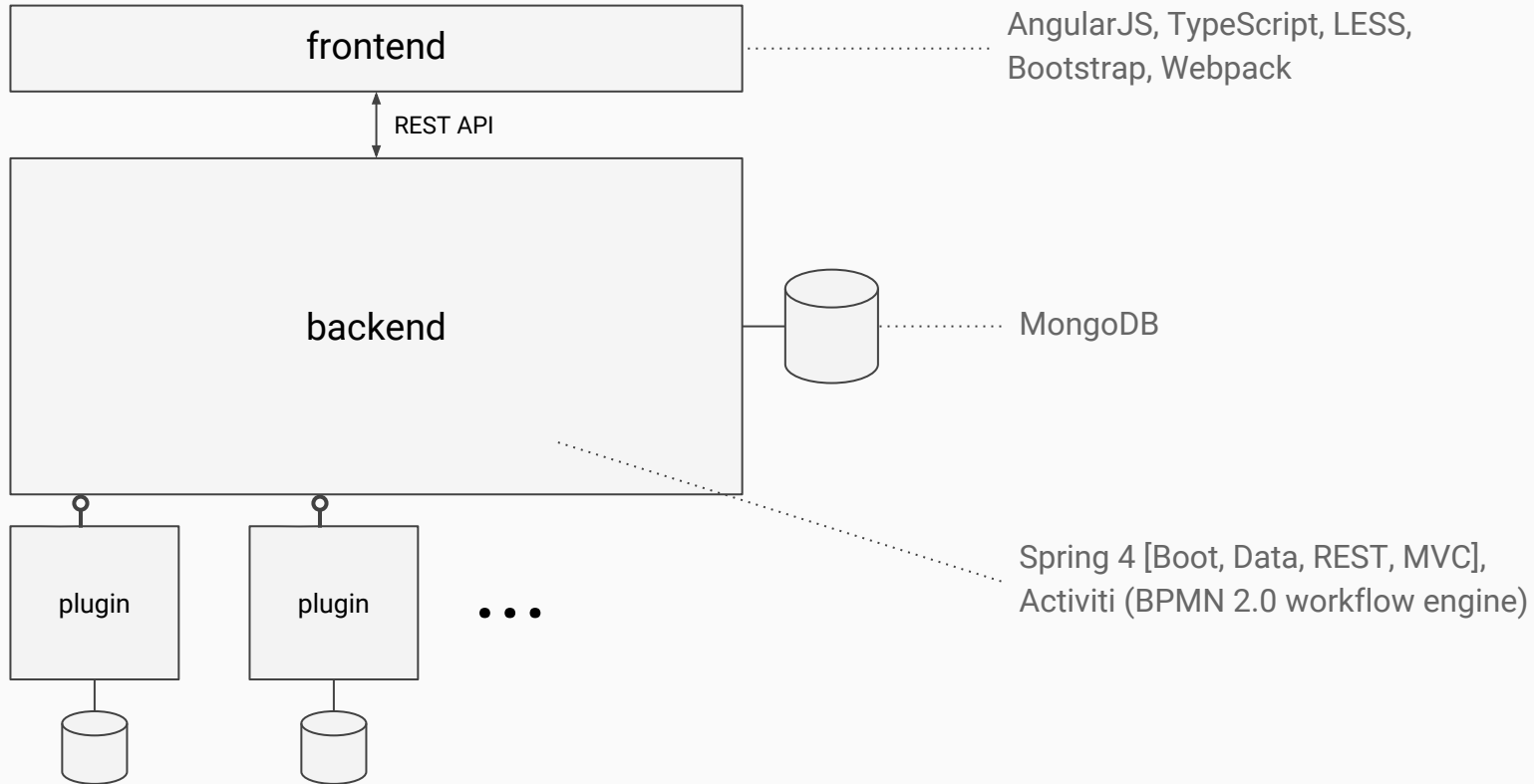- Technology Stack

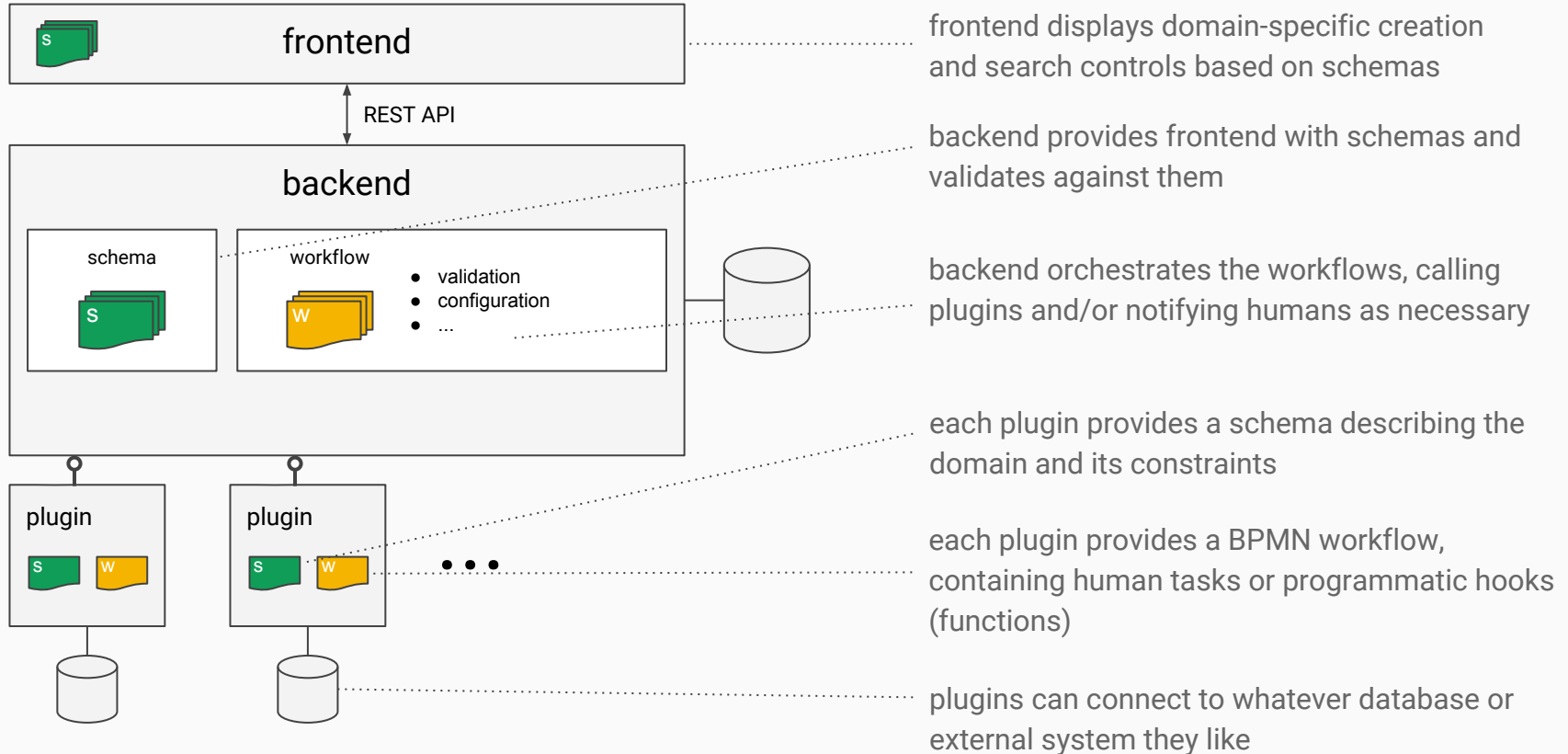- Status and Roadmap

Purpose

Purpose

- Allow both experts and non-experts to enter data for multiple heterogeneous domains/systems via a web application

    - Read/write
    - Sync with external systems
    - Currently mostly monitoring/SCADA systems e.g. TIM, CSAM, WinCC OA

- Cater for domain-specific **workflow** requirements

    - Approval, external sync, testing etc.

- Do not own the data: Connect to existing configuration database(s)

Architecture

Architecture

frontend — AngularJS, TypeScript, LESS, Bootstrap, Webpack

REST API

backend

MongoDB

Spring 4 [Boot, Data, REST, MVC], Activiti (BPMN 2.0 workflow engine)

plugin   plugin   • • •

Architecture

frontend displays domain-specific creation and search controls based on schemas

backend provides frontend with schemas and validates against them

backend orchestrates the workflows, calling plugins and/or notifying humans as necessary

each plugin provides a schema describing the domain and its constraints

each plugin provides a BPMN workflow, containing human tasks or programmatic hooks (functions)

plugins can connect to whatever database or external system they like

frontend

REST API

backend

schema

workflow
- validation
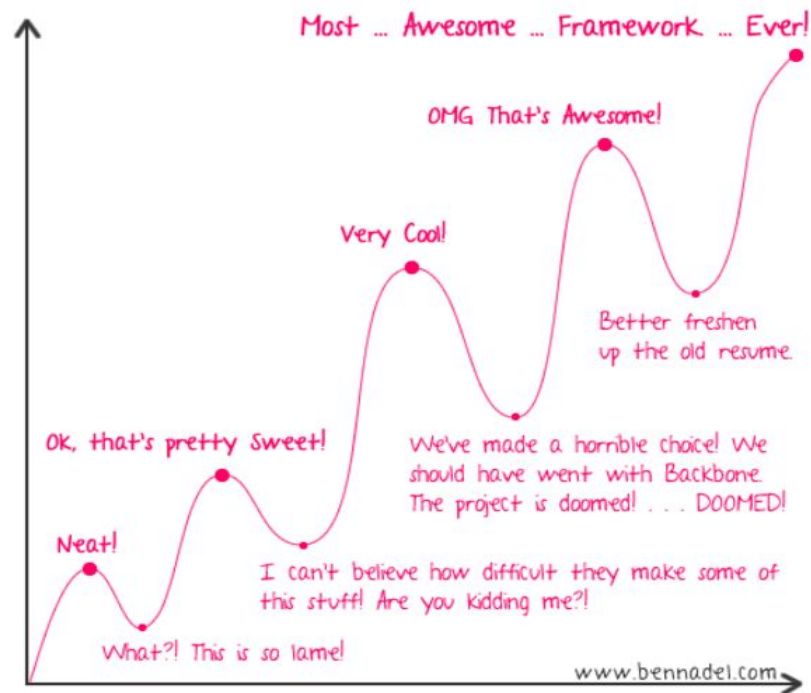- configuration
- ...

plugin

plugin

Technology Stack

AngularJS 1.5

- Enables rapid development and prototyping
- Sound architectural patterns
    - Dependency injection
    - Components, services
- Huge variety of library code available
- Open source

TypeScript

- Type safe (ish)
- Classes, modules, encapsulation, etc.
- Generally makes JavaScript less bad

My Feelings About **AngularJS** Over Time

Technology Choices

MongoDB

- Schemaless, object-oriented datastore allows rapid development
- JSON all the way down
- Replication and sharding out-of-the-box

Activiti

- Fast, lightweight, robust, transactional workflow engine
- In-house knowledge

Problems

3rd-party JavaScript libraries...

- Many, many excellent libraries

- Sometimes it can take a while to find one that fits your exact needs

- Often need to prototype with several

SSL and SSO

- A massive headache until you get it right

- If you can let someone else deal with it, you probably should

Tooling / Deployment

Tooling / Deployment

Frontend

- NPM
- Webpack

Backend

- Gradle
- Spring Boot (embedded web server)

Application runs as a standalone Java process with Apache HTTPD in front

Monitored via wreboot / DIAMON

Current Status and Roadmap

In production since January 2016

30-40 users

Roadmap for v0.2.x (end of the year):

- Searching, updating and deleting existing data
- Frontend extensibility (via dynamic module loading)

Thank you!

any questions?