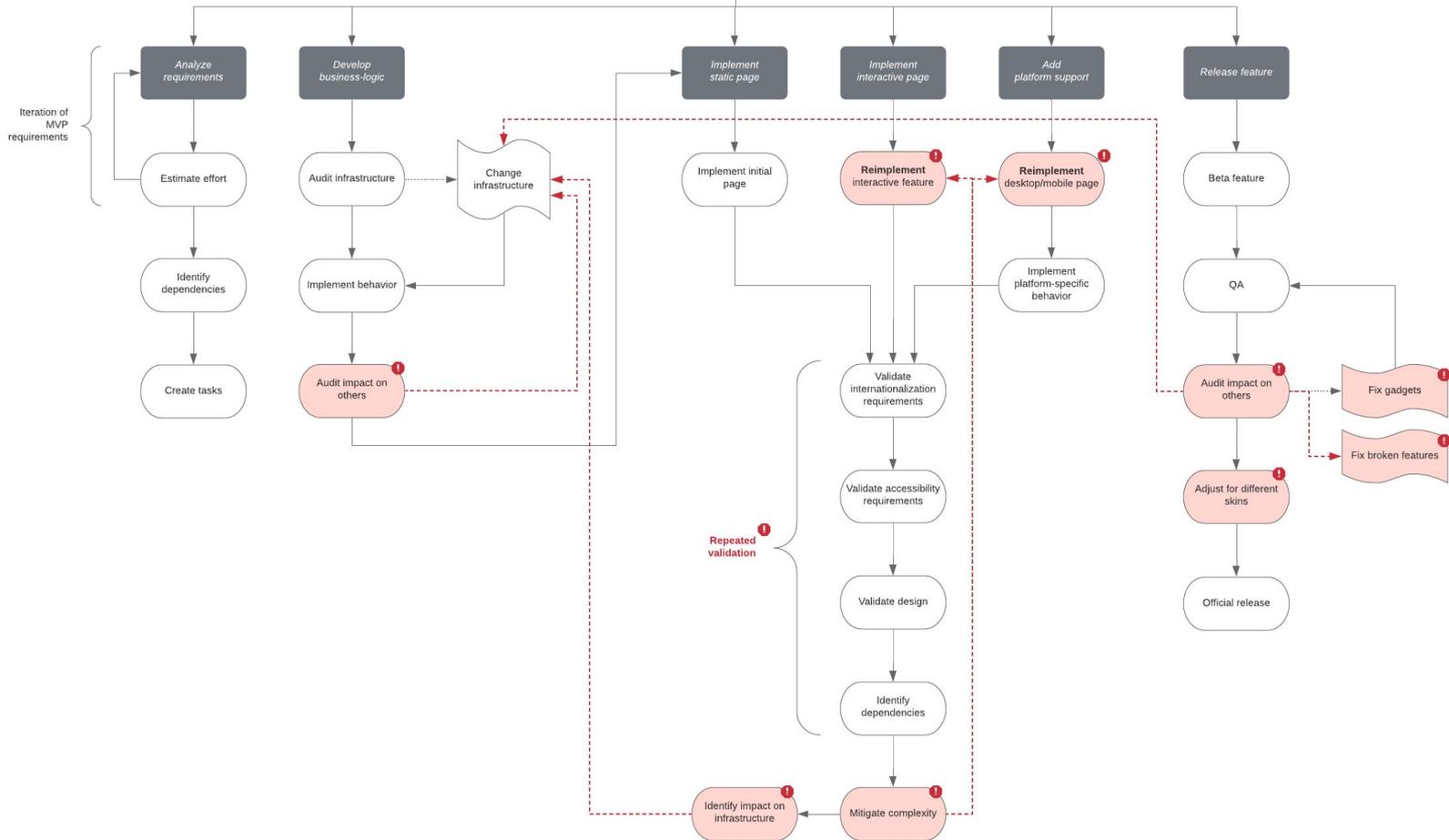


Proposed Metrics

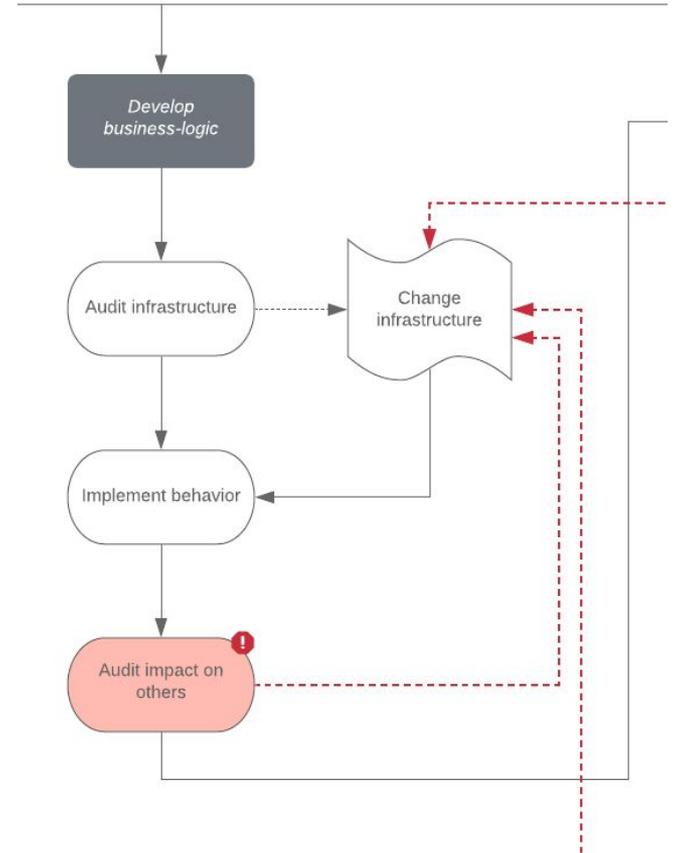
- Developer satisfaction over time
- Product manager and Designer satisfaction over time.
 - Was Product able to deliver on visions of modern UIs?
 - Were designers able to see a design roll-out as they envisioned?
- SUS score on a product level
 - Collect baseline now and compare after overhaul

Current developer workflow



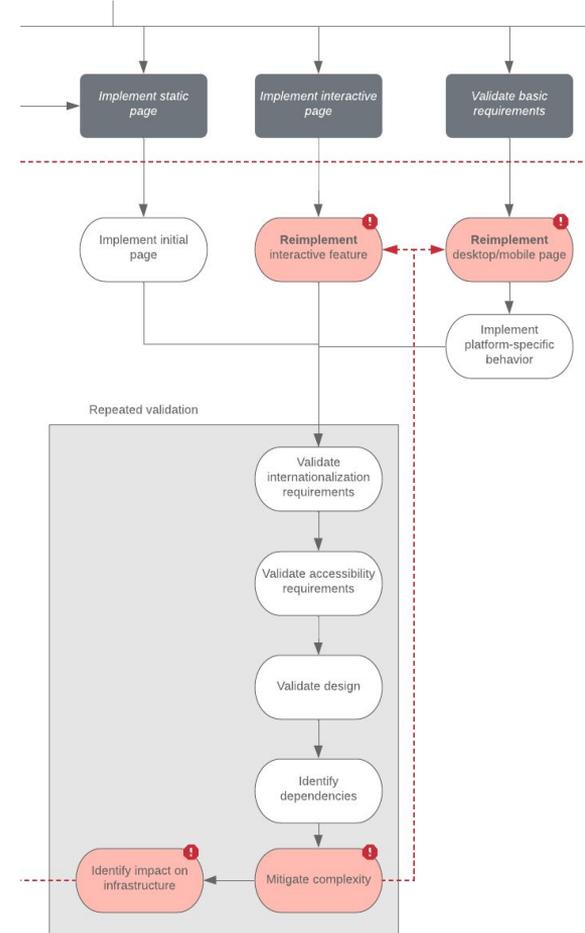
Current developer workflow

- Entangled code means constantly checking impact on other pieces
- Even small features often require infrastructure change



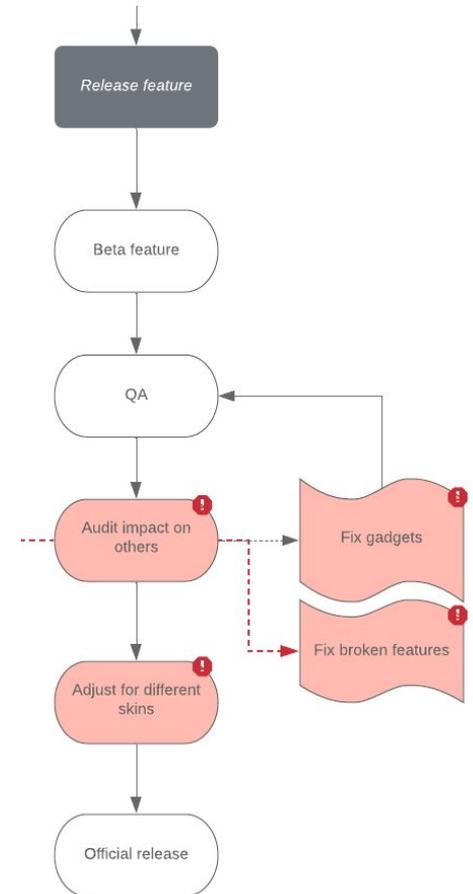
Current developer workflow

- Duplication of code (or intentionally avoiding duplication in agreement with product at the expense of user experience)
- Repeated validation for each platform and between backend and frontend

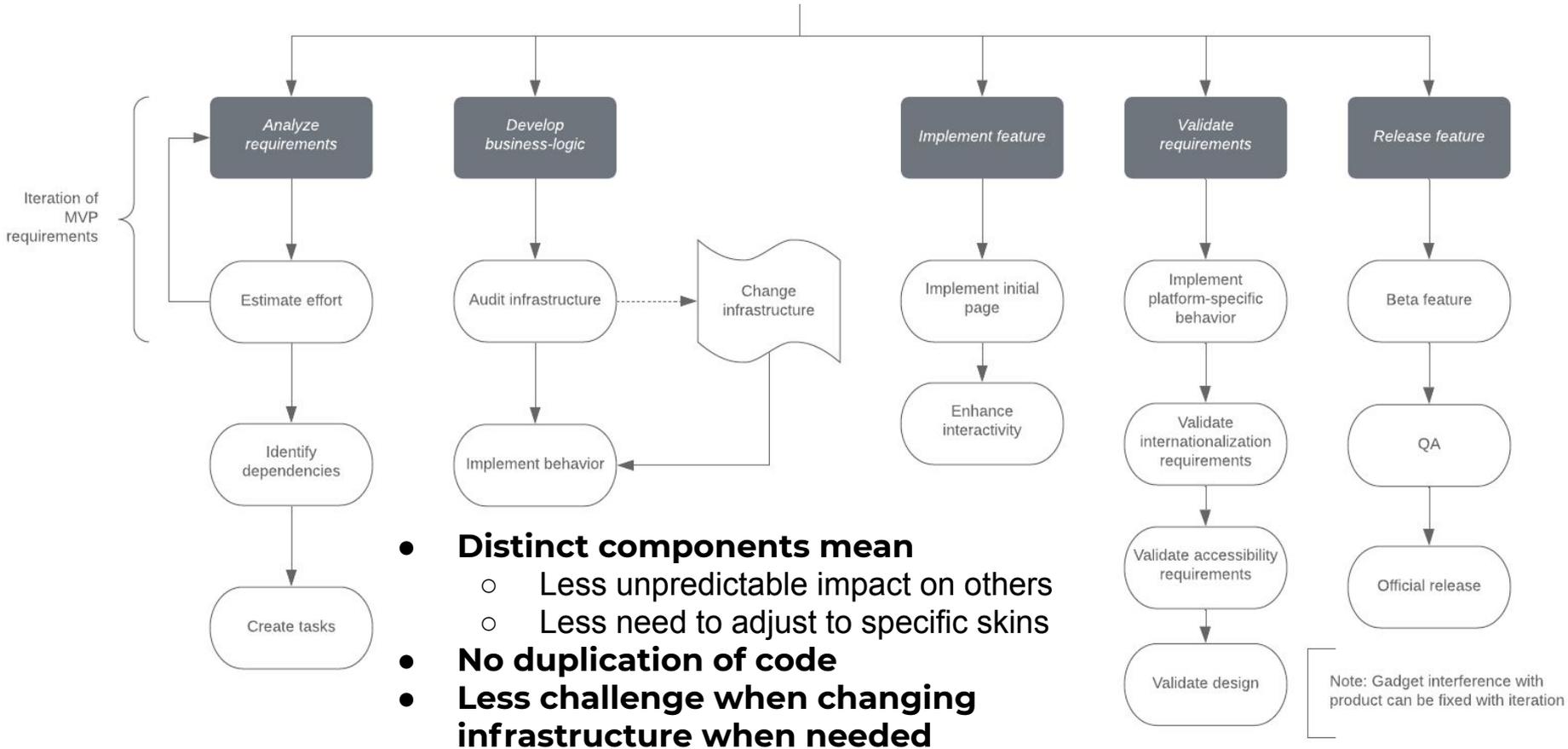


Current developer workflow

- Risking impacting unrelated features even on product release
- Adjustment for skins and gadgets



Ideal developer workflow



Data exchange

In decoupling, the contract between backend and frontend is data.

There needs to be a definition of data:

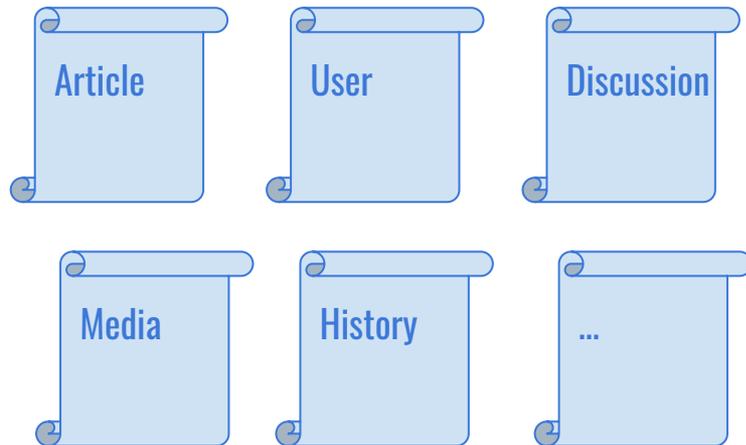
- **Formal**
- **Validatable**
- **Architecture-independent**
- **Versioned**

Data exchange

In decoupling, the contract between backend and frontend is data.

There needs to be a definition of data:

- **Formal**
- **Validatable**
- **Architecture-independent**
- **Versioned**



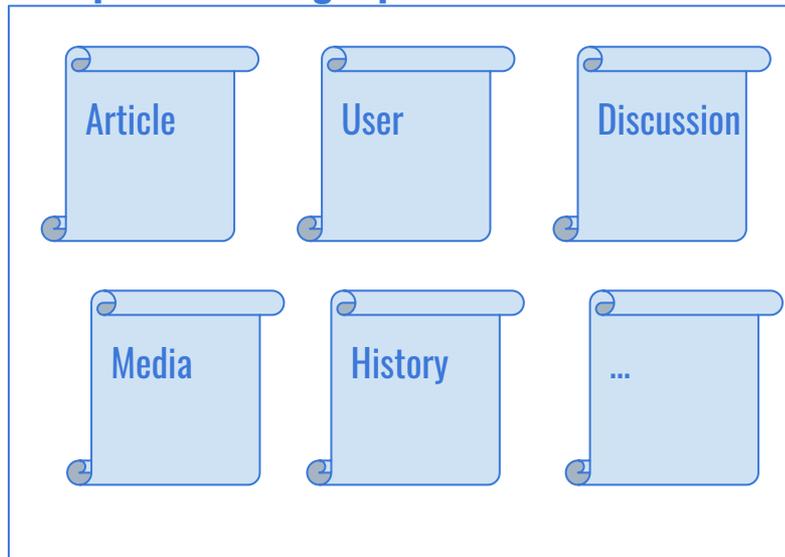
Data exchange

In decoupling, the contract between backend and frontend is data.

There needs to be a definition of data:

- **Formal**
- **Validatable**
- **Architecture-independent**
- **Versioned**

Wikipedia Knowledge Specification

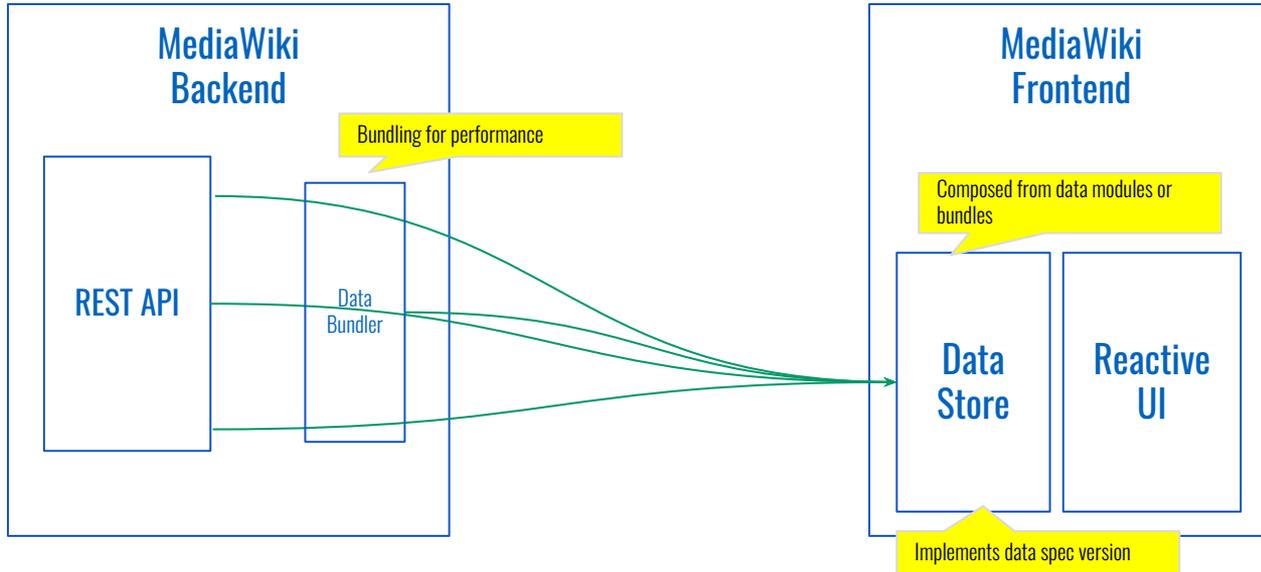


Data exchange

Rationale

- From presentation dependent Wiki content to Structured data bundles.
- For developers, a well-defined data model helps to reason about, debug, manipulate, and build user interface and interactions.
- Facilitate diverse frontend experience development by anybody who wants to present Wikipedia content in a suitable form.

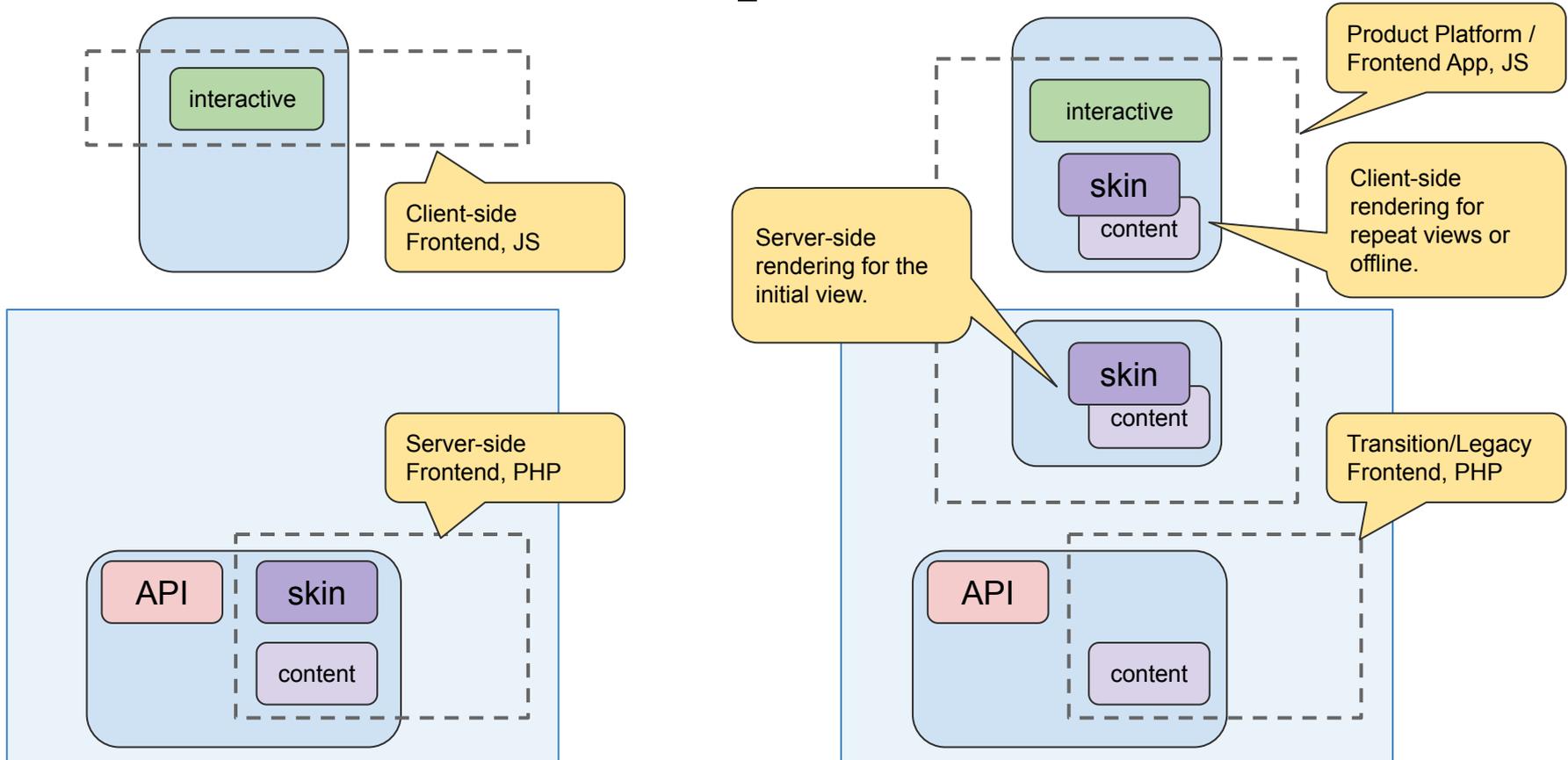
Data exchange



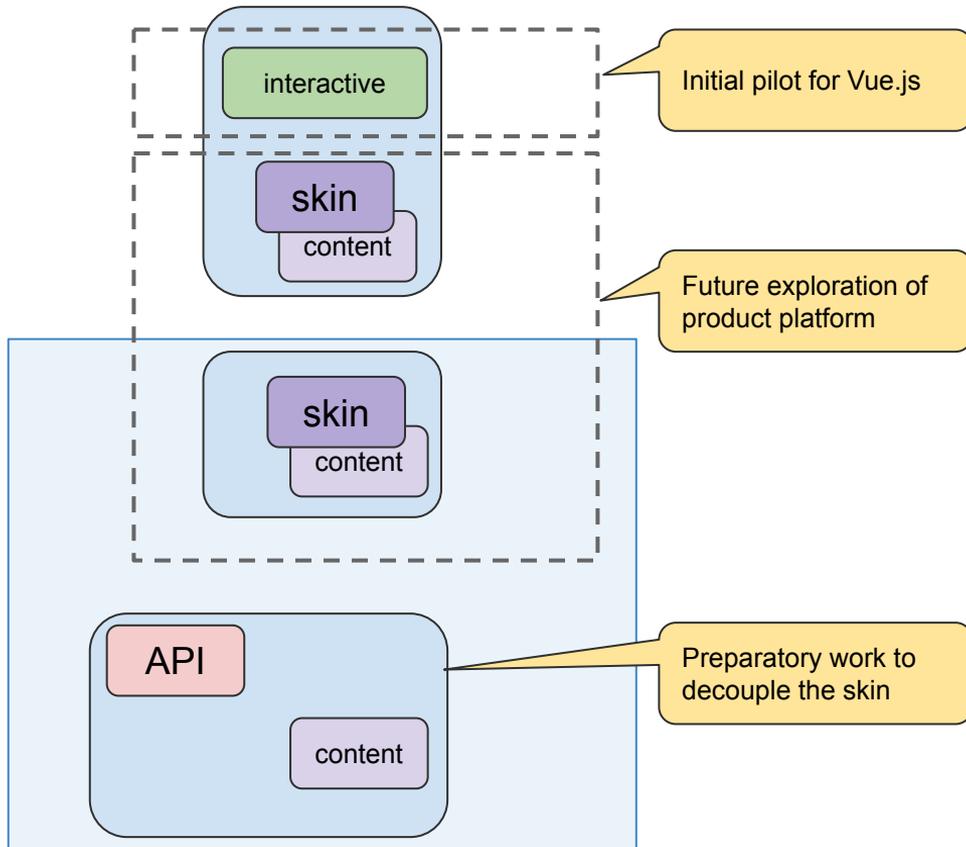
Possible Product Platform Architecture



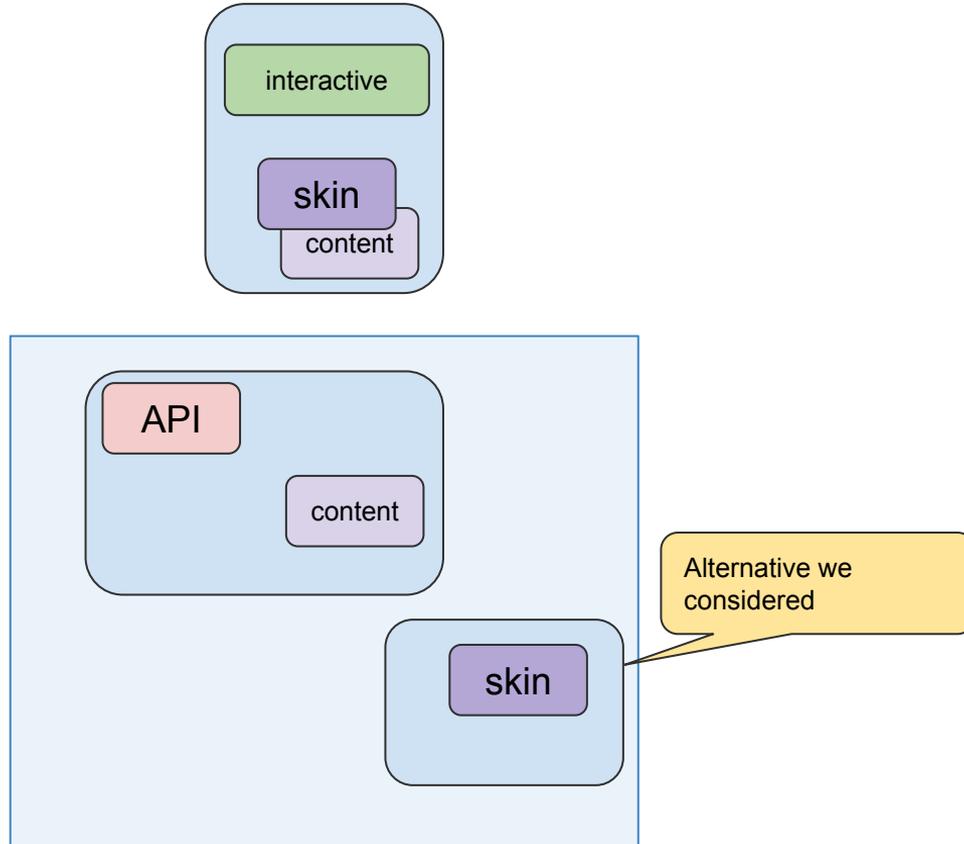
Platform Exploration



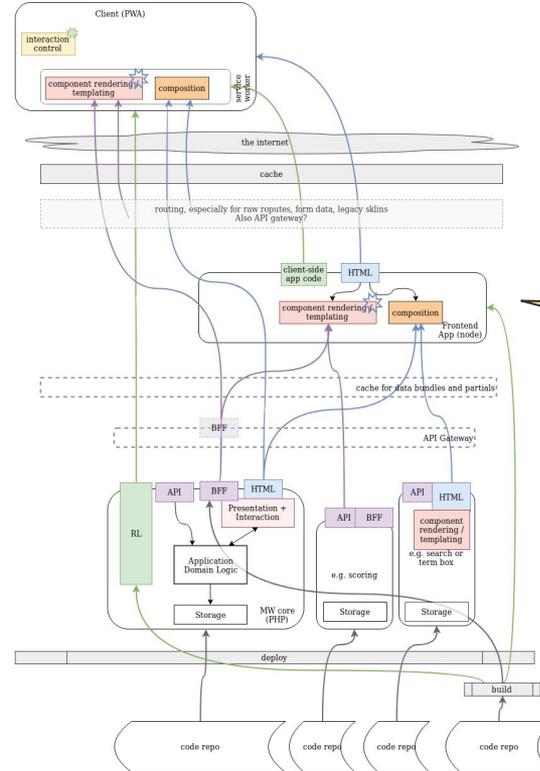
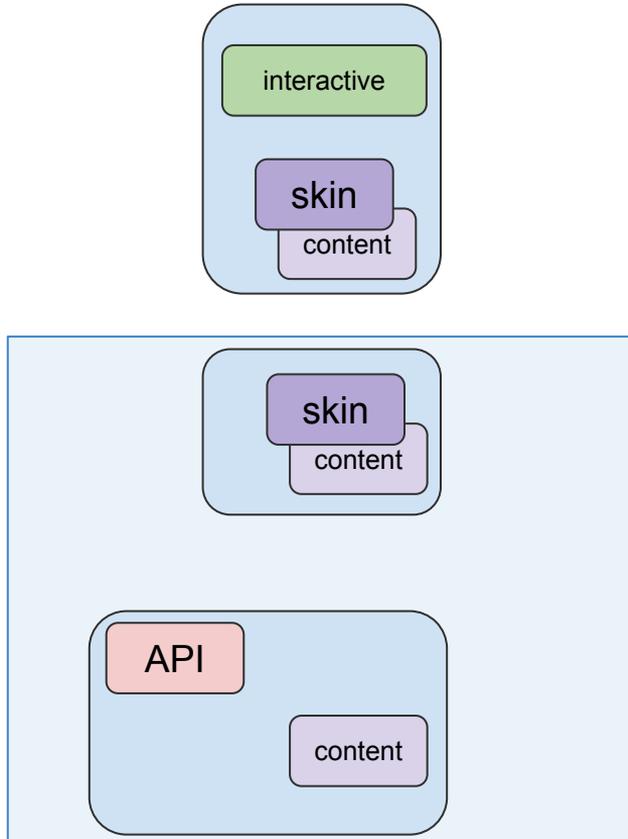
Platform Exploration



Platform Exploration

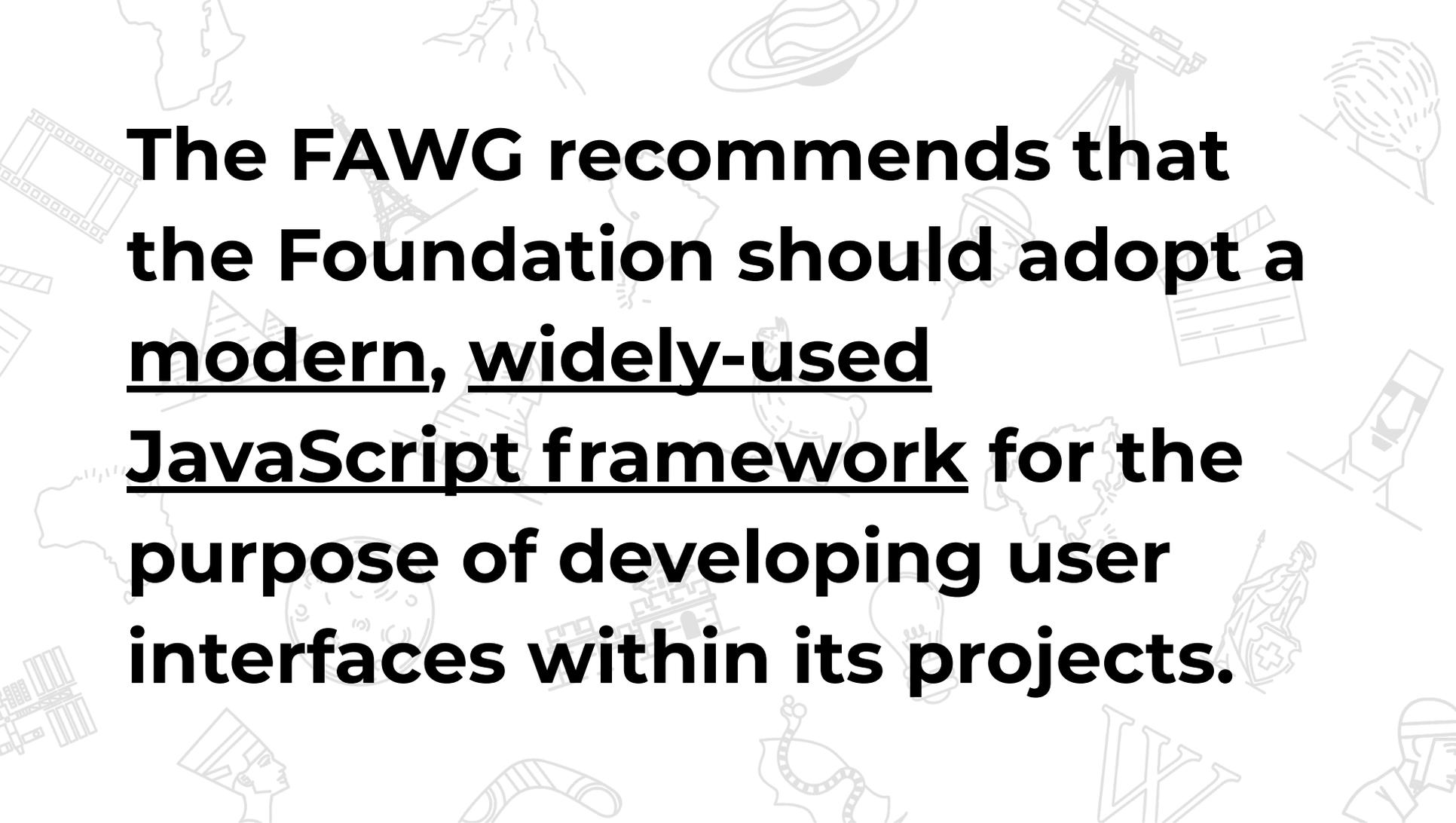


Platform Exploration



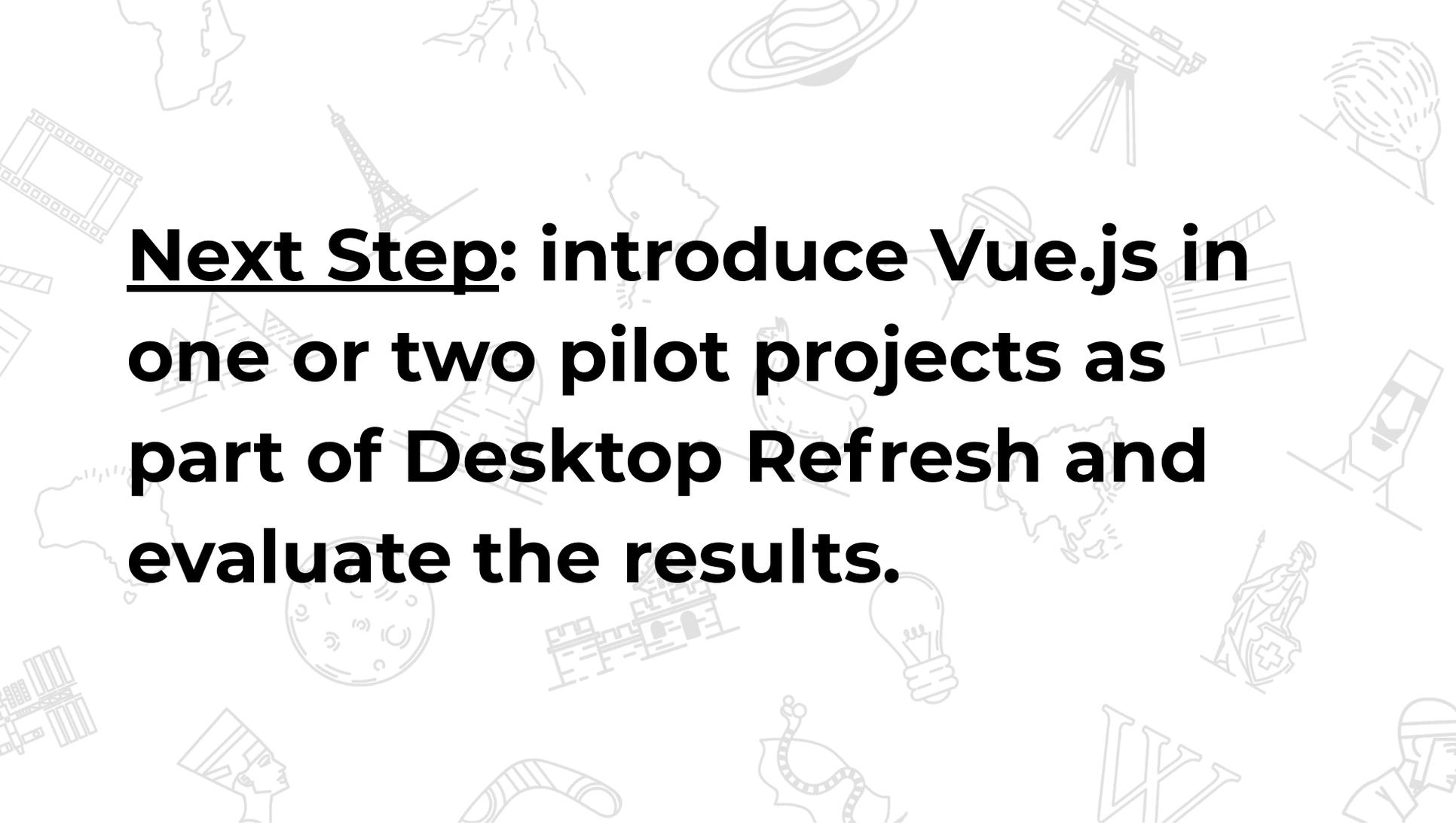
Introducing Vue

The background is a solid green color with a pattern of white line-art icons. These icons represent a wide variety of topics: science (telescope, planet Saturn, microscope, lightbulb), history and culture (Eiffel Tower, Great Pyramids, Sphinx, Statue of Liberty, ancient Egyptian head, film strip), geography (maps of Africa, Australia, and Asia), technology (satellite, smartphone), and general concepts (mountain, film camera, clapperboard, film reel, film strip, film strip, film strip).



The FAWG recommends that the Foundation should adopt a modern, widely-used JavaScript framework for the purpose of developing user interfaces within its projects.

- **Many ideas explored by the FAWG would represent major changes to MediaWiki, and would take time to implement**
- **Vue.js can help pave the way for some of these bigger changes, but it is also a tool we can start using now, within our existing infrastructure**
- **Potential benefits include: better developer experience, faster development of new features, lower barriers to contribution, and fewer UI bugs**



Next Step: introduce Vue.js in one or two pilot projects as part of Desktop Refresh and evaluate the results.

The Long Road Ahead

- One of the deeper outcomes of the FAWG is that our **ability to improve** frontend developer experiences through frontend changes is **limited by the architecture of our backend software and APIs**.
- **The group has uniquely facilitated valuable conversations between key parties** about these deeper issues, but we're still in the early stages of arriving at a shared language and vision for how we should migrate towards a more-supportive backend architecture.
- **Continuing these conversations is vital** as we iteratively converge on novel architectural solutions while facing the complex integration and migration challenges of our environment.

Product platform architecture

- Structuring content
- Moving presentation and interaction logic into the frontend
- Rendering pages/parts of pages
- Creating choreography
- Internationalisation
- And more... on essential components

