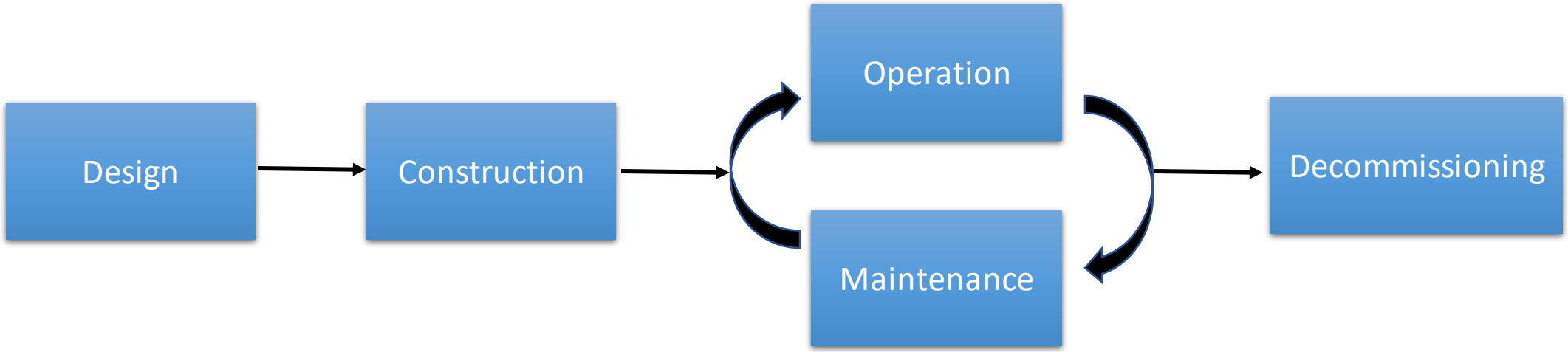
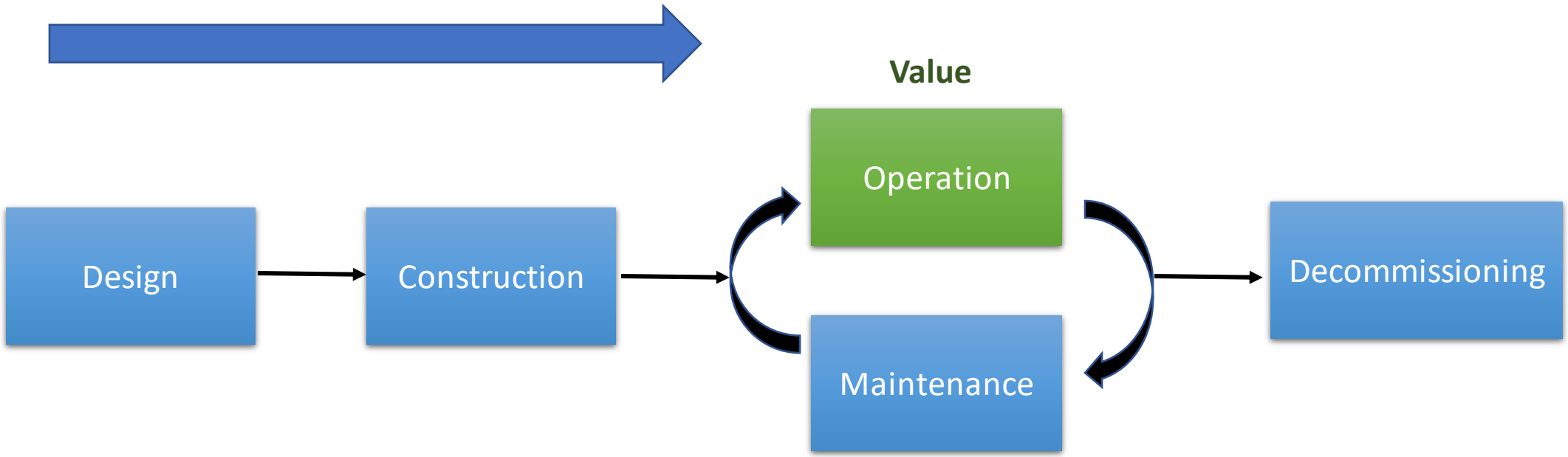
Question2

Trip Dacus Interview for Oklo Inc.

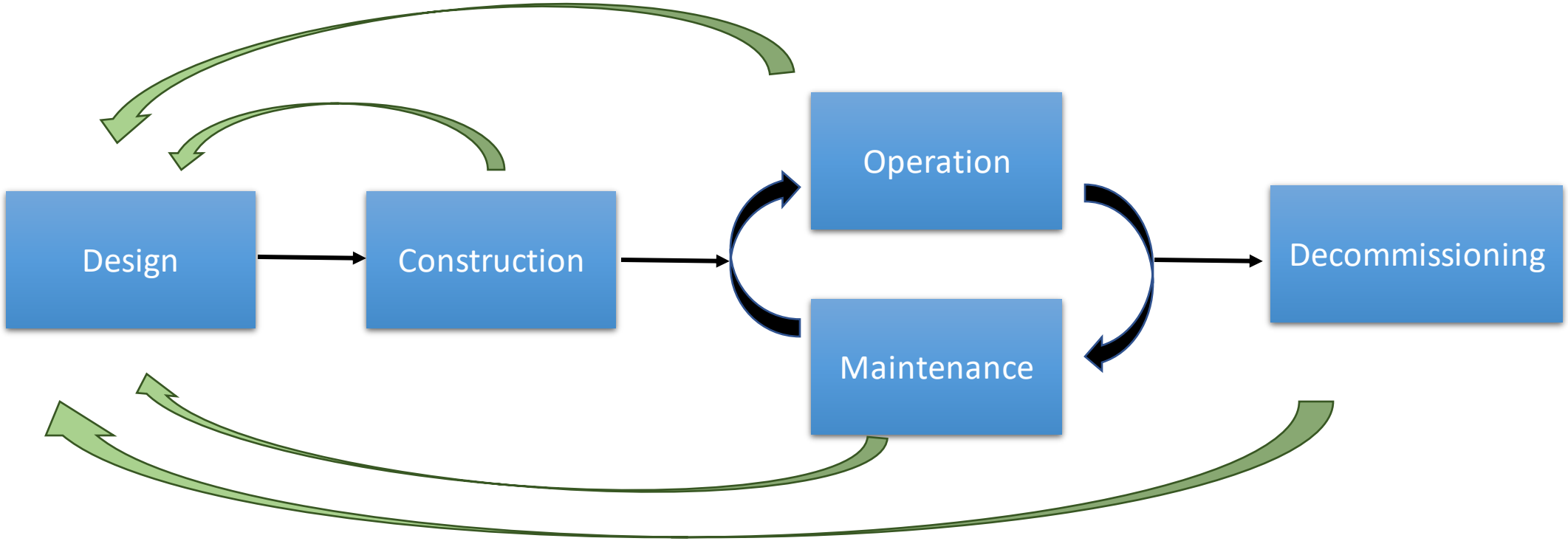
Overall Process



Maximize Throughput to deliver value to the customer



Design feedback at every stage of the process

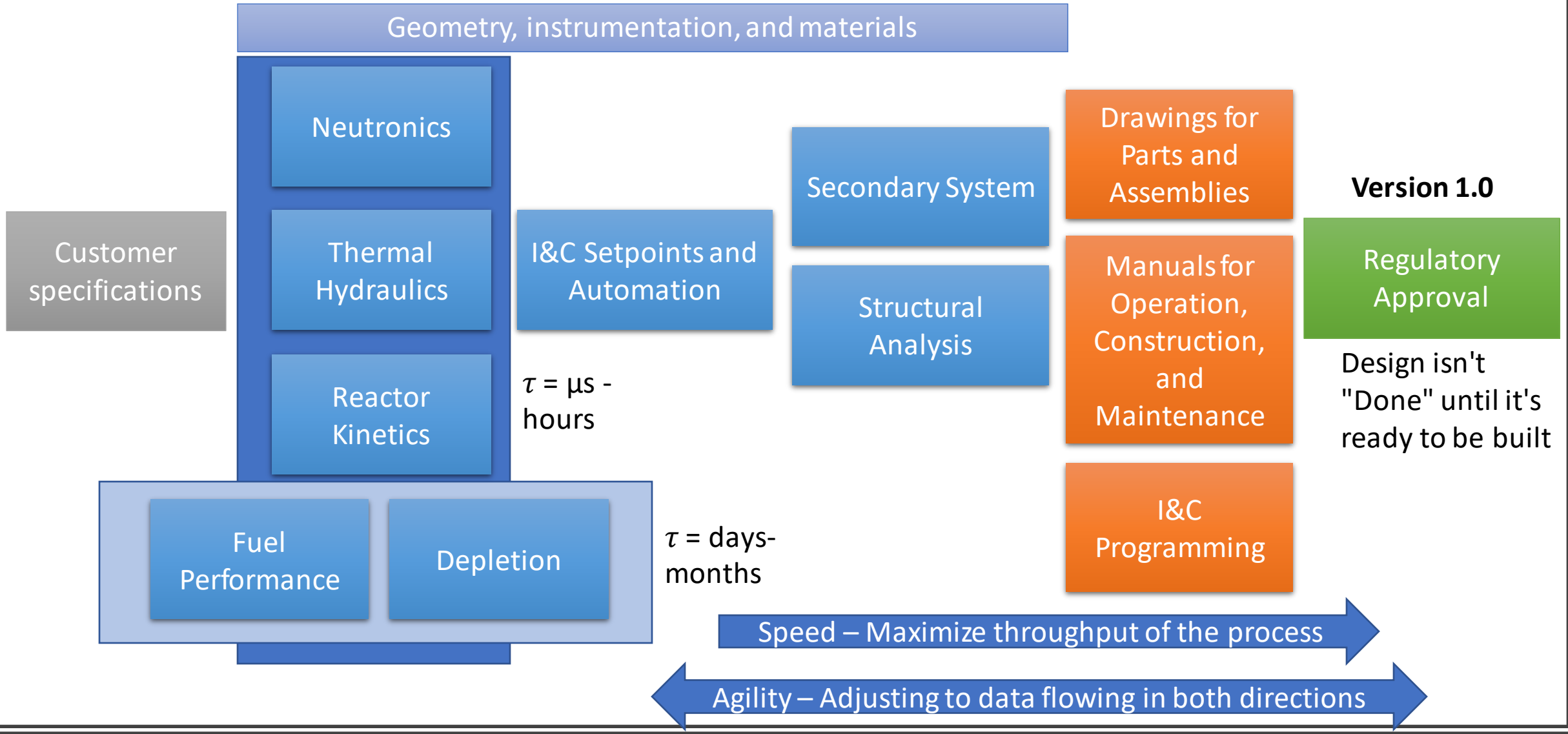


Agility

Changing direction at fast speed

Design versioning process must enable rapid throughput and iteration while maintaining accuracy and reproducibility

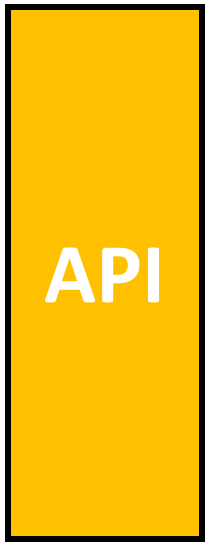
Design Process



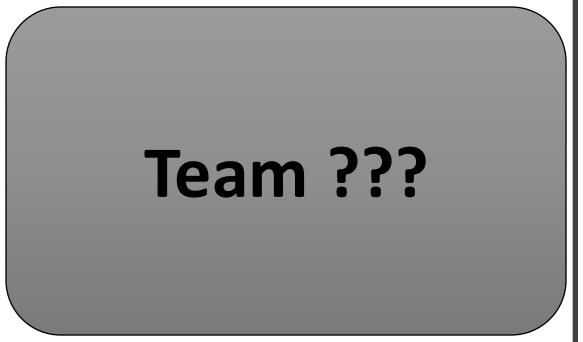
Asynchronisity

- Design sub-process independence enables flexibility and encapsulation
  - Makes design process easier to think with
- Design process "APIs" and self-organization
- Less brittle design process
- Corollary to supply chain logistics and Continuous Integration

Version 42.42



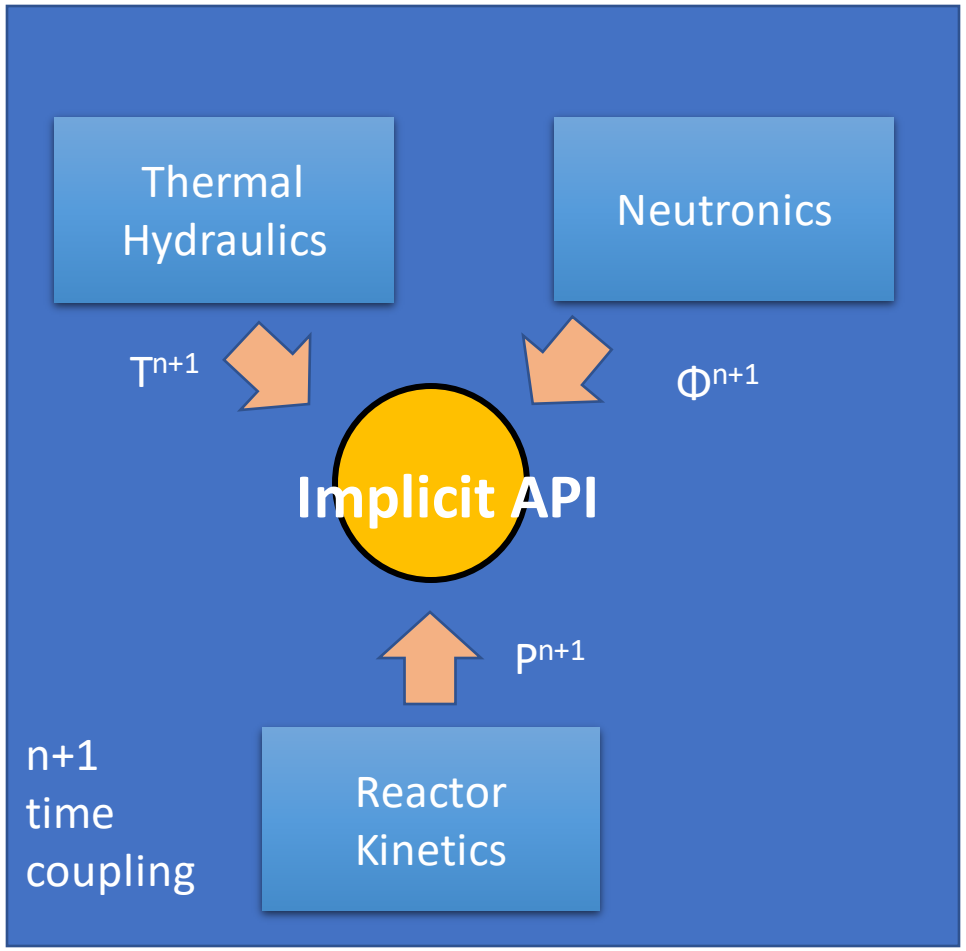
Version 1.2.1



Version 9999

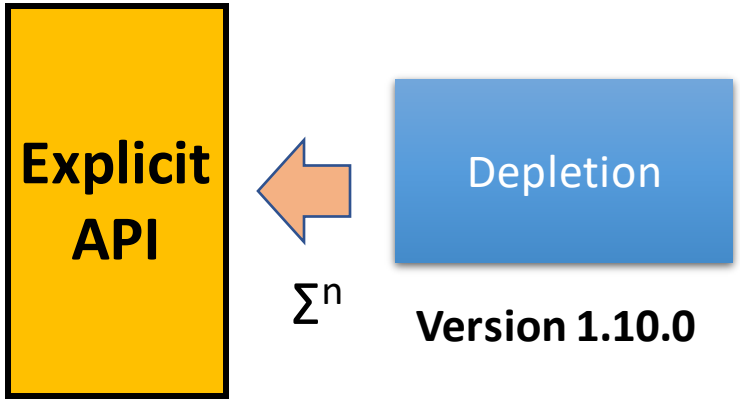
Voltron not worried as long as the API is stable

Versioning for Explicit and Implicit Processes



Version 4.3.2

- Not everything can by asynchronous
- Coupling improves accuracy but increases stiffness of the system

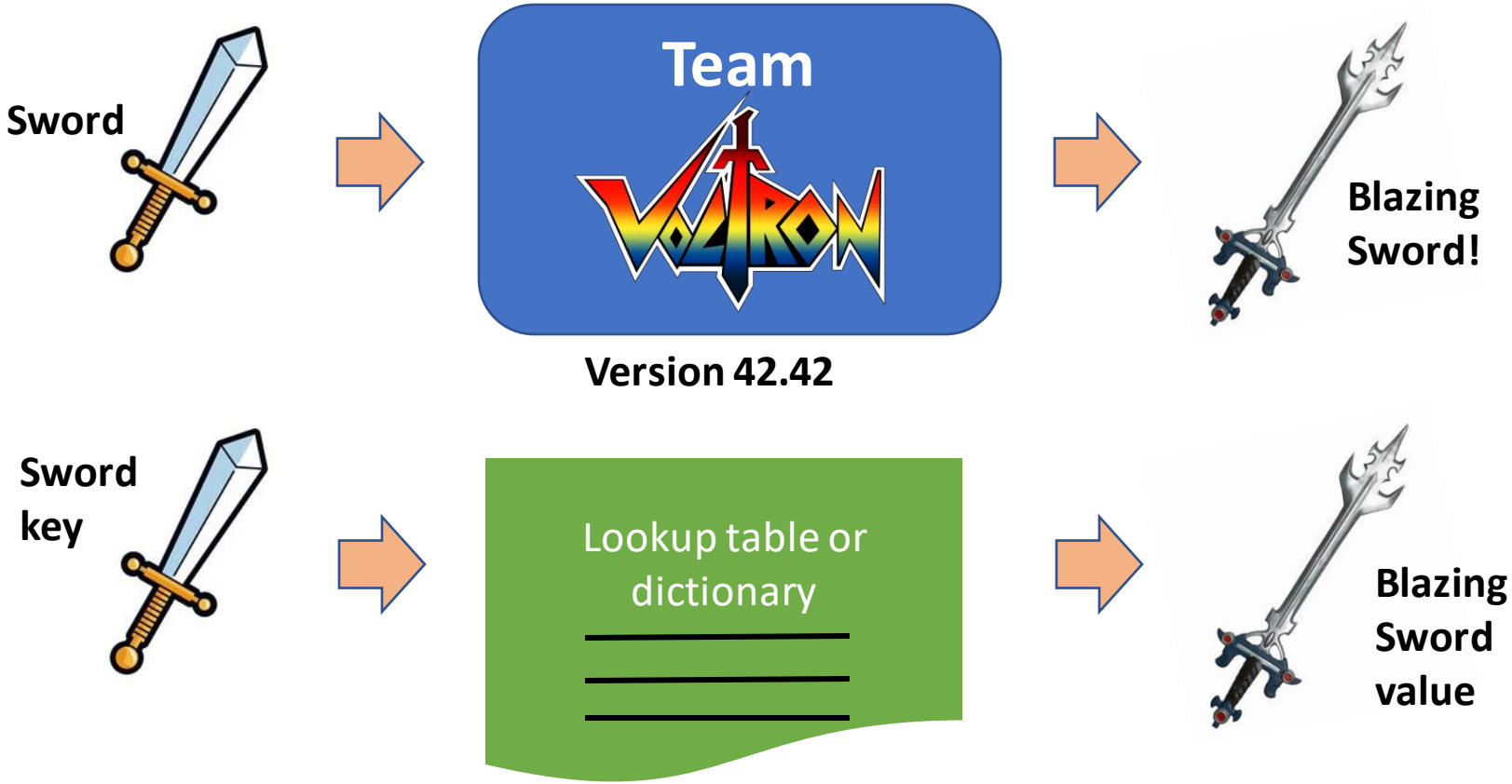


Version 1.10.0

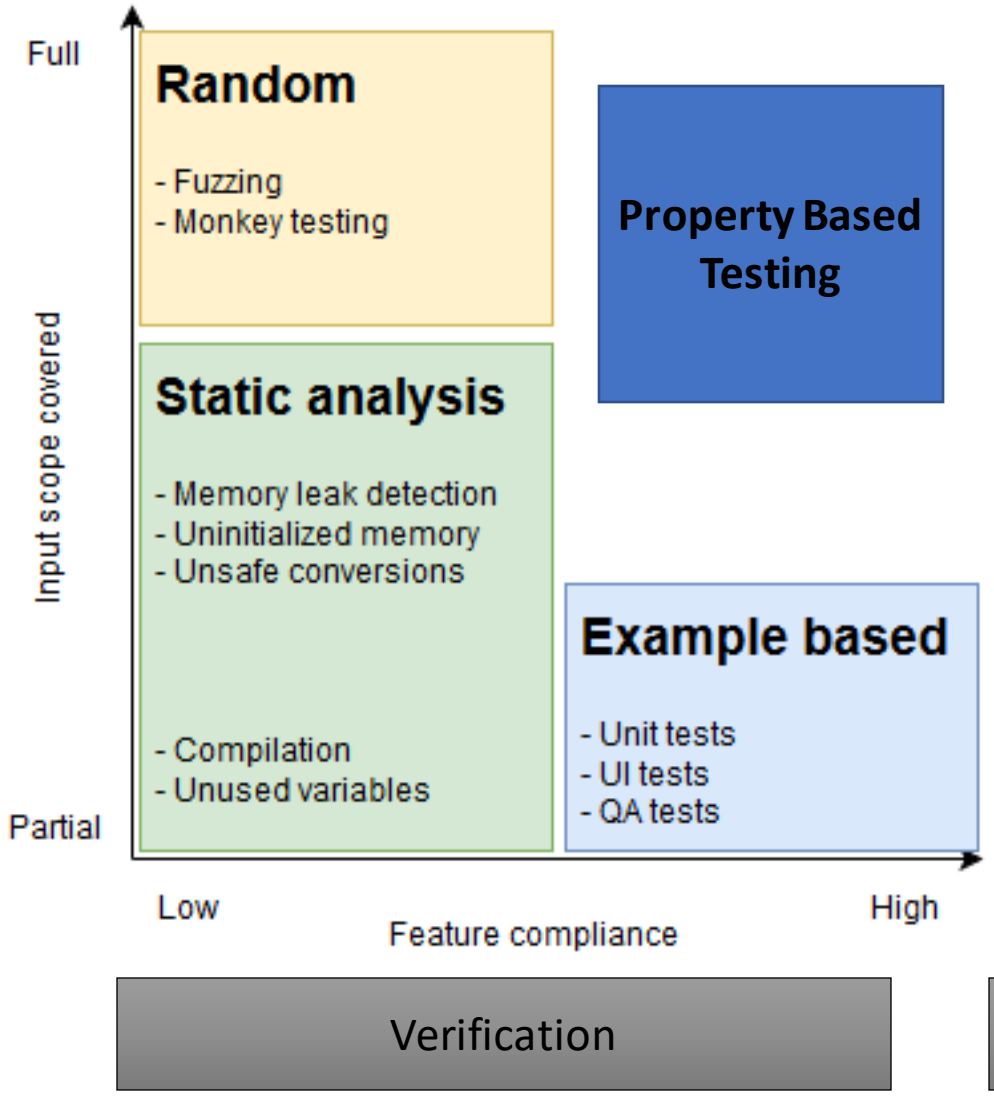
n time coupling

Duality of Code and Data

- If processes are "pure," can substitute code and data
- Don't have to recompute values for successive iterations



Automated Testing and Human Checking



- Maximize automation and early-error detection
  - "A bug left overnight becomes a feature in the morning"
- Engineers must still make a judgement
  - Lightweight review process
  - Automatically generate what the engineer needs to approve a process change

Theory of Constraints

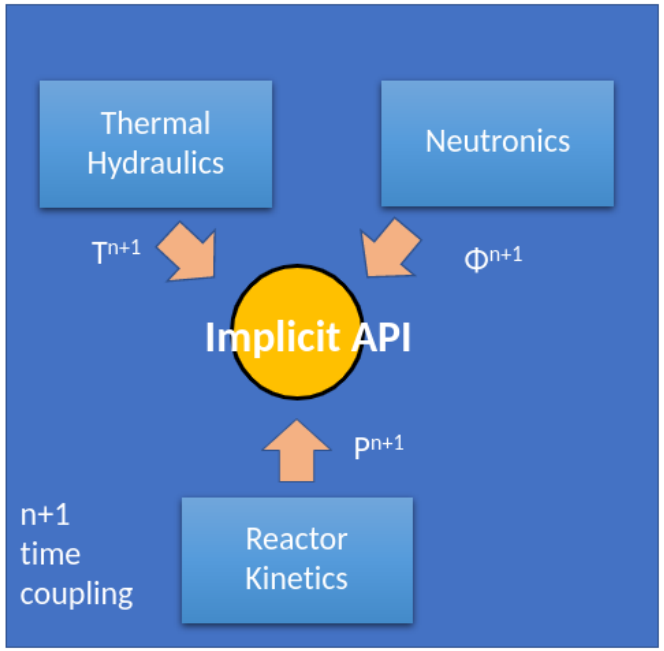
"A system of local optimums is a very inefficient system" - Eliyahu Goldratt

Throughput of a system is dictated by the throughput of the constraint of the system as a whole

Maximizing Throughput by Focusing on the Constraint

- Where is the constraint in the design process?
- Does migrating to the cloud improve the throughput of the constraint?

Throughput Rate ??



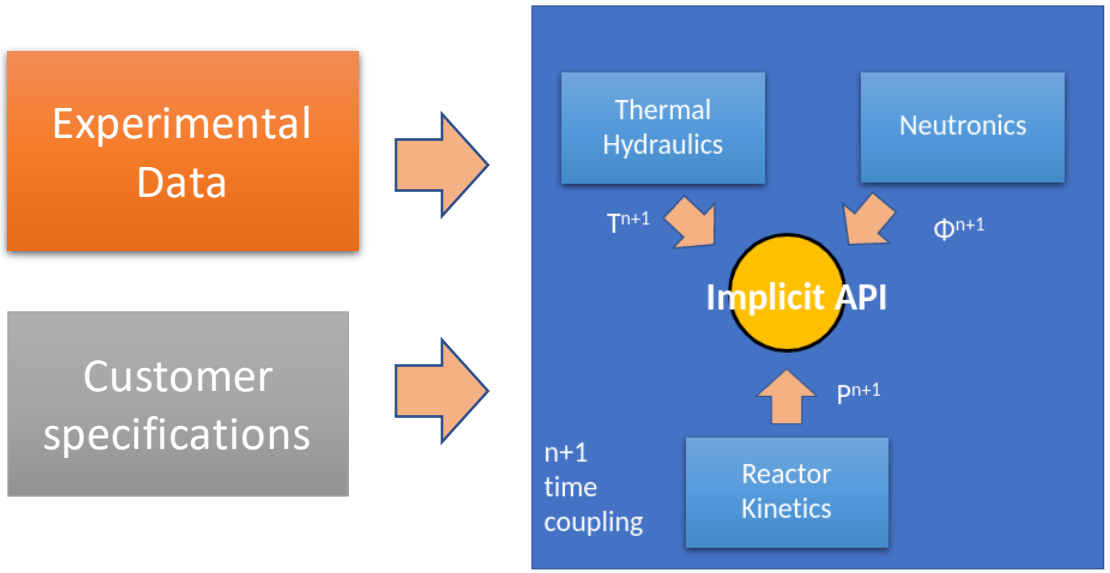
Throughput Rate ??

Regulatory Approval

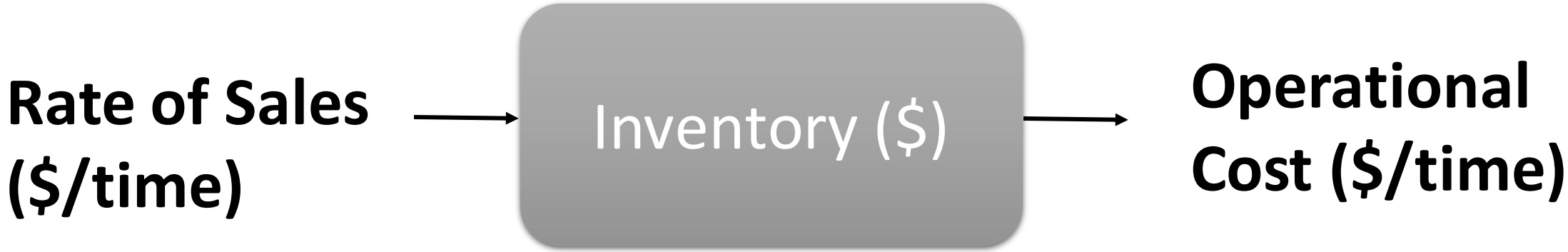
Data to Provide Focus

Where do you need additional experimental data?

- Technical "constraint" of the design process
  - Where a design feature limits performance to customer requirements
  - What is the customer's unrefusable offer?
- Change data capture for reproducibility



Throughput Accounting



Cloud Migration

Pros

- Cloud was designed for
  - Asynchronisity
  - Scalability
  - Change data capture
- Don't have to maintain physical hardware
- No lead time for new infrastructure (hardware/software)
- Reliability?
- Dedicated support

Cons

- Requires investment
- Increased cost?
- Implicit calculations require parallelization to scale
- Dependence on commercial vendor

Does it increase **Throughput** or **Agility** ?

Redundancy and Reproducibility

Can be established using Change Data Capture for any system

- git version control for source code
- Publish/Subscribe for experimental data
- Publish/Subscribe for explicit coupling interfaces
- Kubernetes to manage microservices