Digital Twin as a Service Software Platform

Prasad Talasila

prasad.talasila@ece.au.dk

Acknowledgments

John Fitzgerald, Claudio Gomes and Peter Gorm Larsen (Editors), The Engineering of Digital Twins (book draft), January 2024.

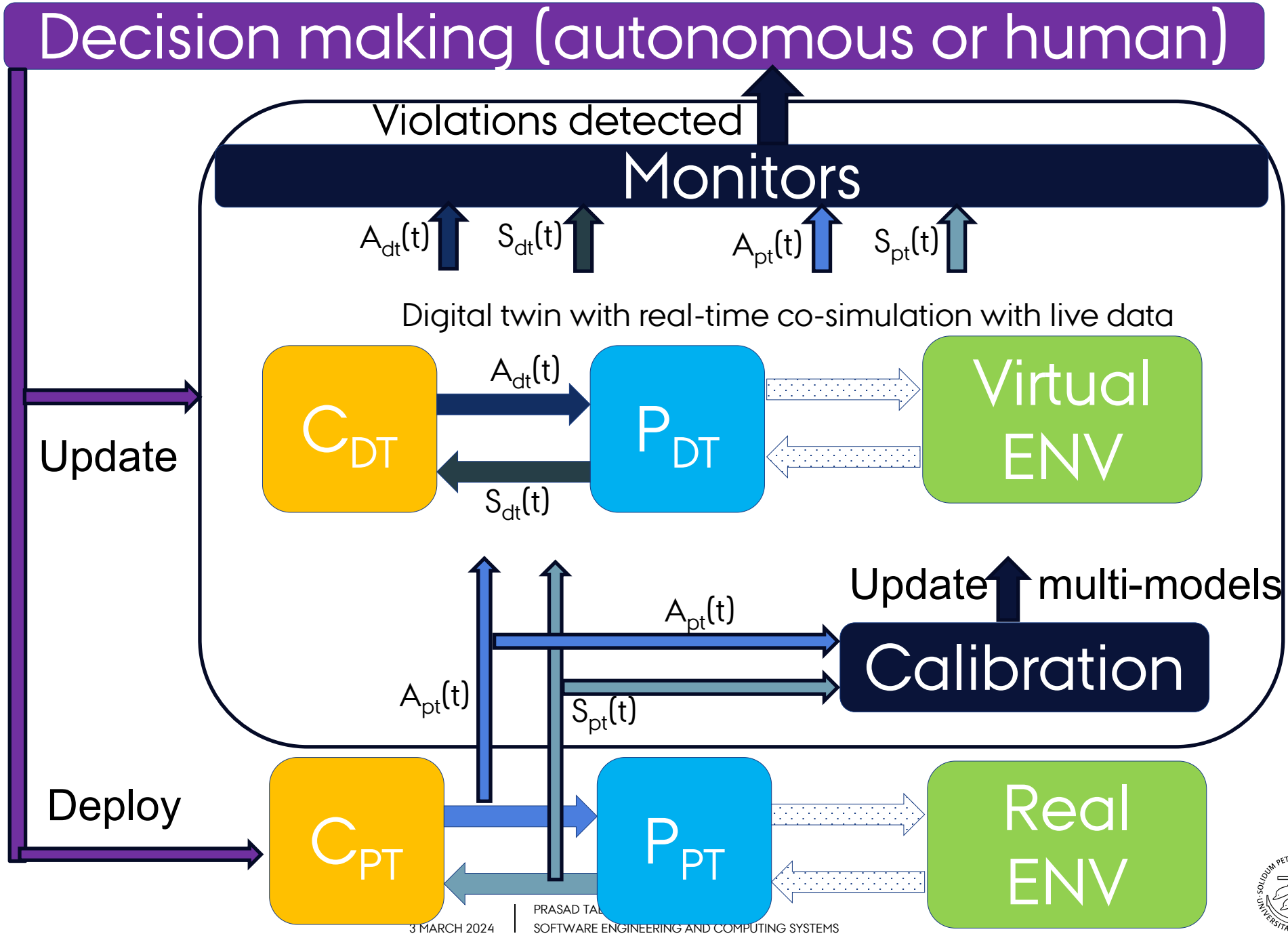
PRESENTATION OUTLINE

1) What is inside a Digital Twin?
2) Who are Users?
3) What are requirements for Digital Twin Platforms?
4) One viable system architecture
5) What is the implementation status?
6) What is to come later?
7) How can you contribute?

PRESENTATION OUTLINE

1) What is inside a Digital Twin?
2) Who are Users?
3) What are requirements for Digital Twin Platforms?
4) One viable system architecture
5) What is the implementation status?
6) What is to come later?
7) How can you contribute?

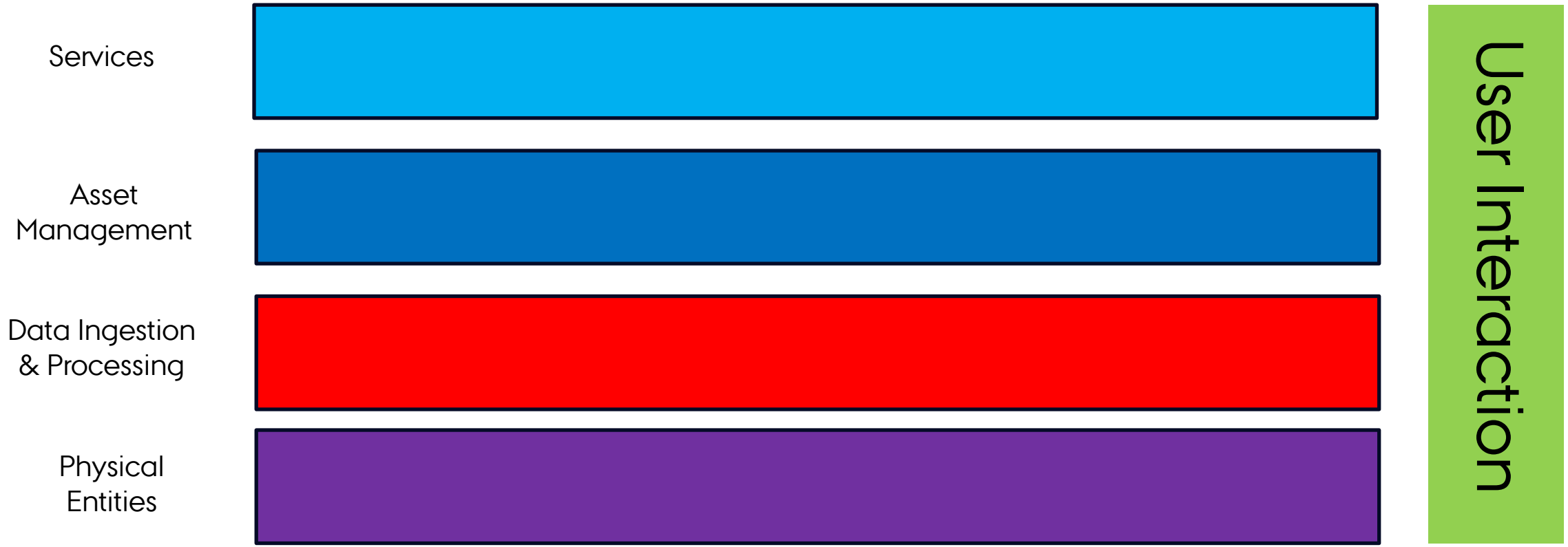
Schematic View of A Digital Twin



C – Cyber P – Physical
DT – Digital Twin
PT – Physical Twin

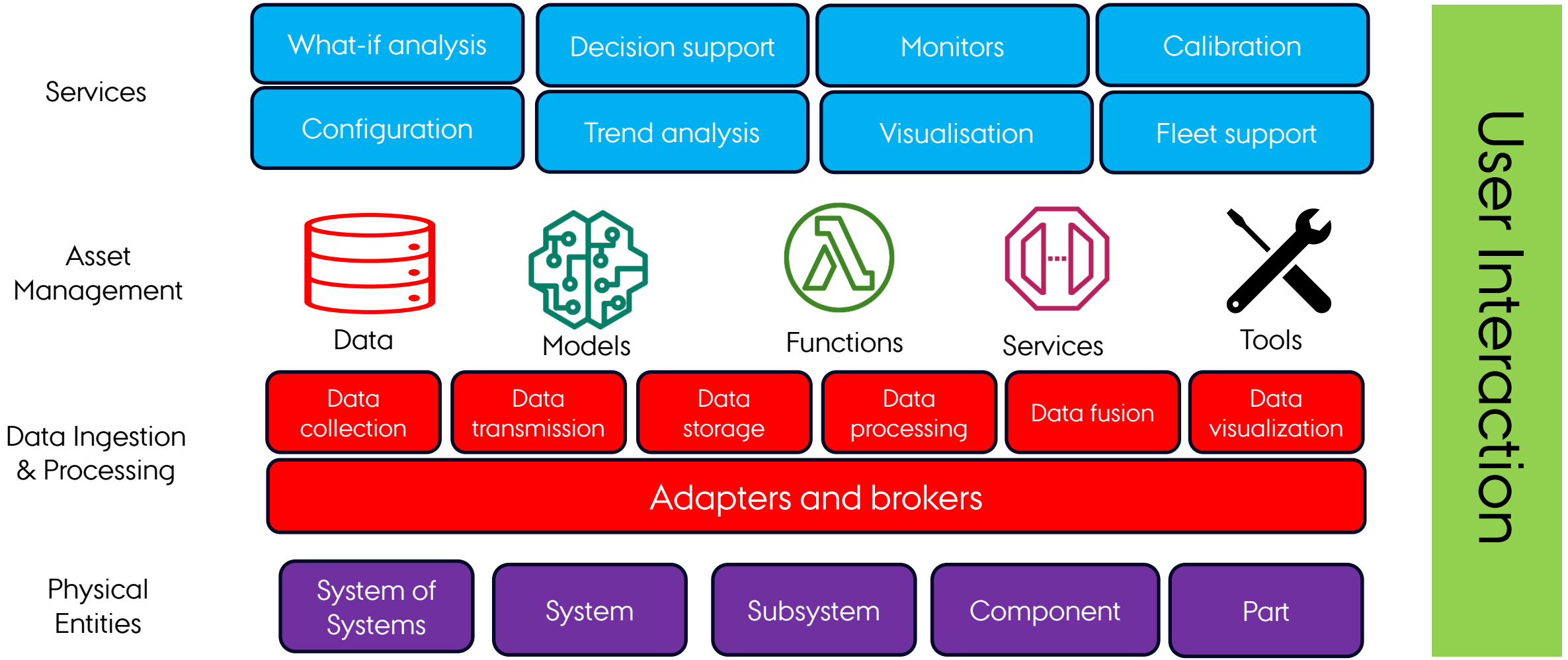
Ref: Feng, H., Gomes, C., Thule, C., Lausdahl, K., Sandberg, M., & Larsen, P. G. (2021). The incubator case study for digital twin engineering. *arXiv preprint arXiv:2102.10390*.

DIGITAL TWIN LAYERS: A PROPOSAL



NOTE: This is not a strictly layered architecture

WHAT IS INSIDE THE DIGITAL TWIN LAYERS?



NOTE: This is not a strictly layered architecture

PRESENTATION OUTLINE

1) What is inside a Digital Twin?
2) **Who are Users?**
3) What are requirements for Digital Twin Platforms?
4) One viable system architecture
5) What is the implementation status?
6) What is to come later?
7) How can you contribute?

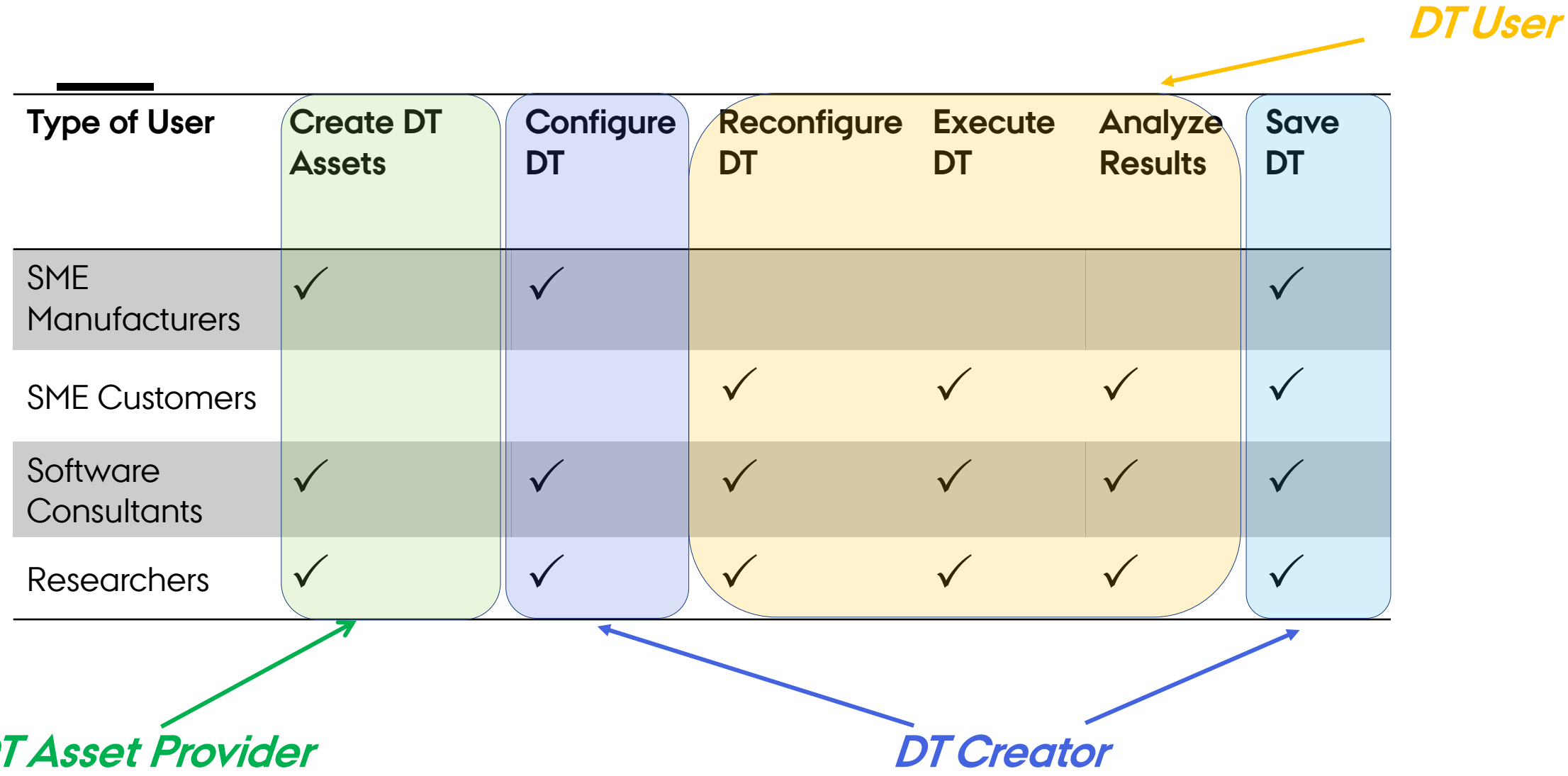
WHO ARE THE USERS?

Type of User	Create DT Assets	Configure DT	Reconfigure DT	Execute DT	Analyze Results	Save DT*
SME Manufacturers	✓	✓				✓
SME Customers			✓	✓	✓	✓
Software Consultants	✓	✓	✓	✓	✓	✓
Researchers	✓	✓	✓	✓	✓	✓

Personas / Roles

*Save DT – Save a running instance of a DT

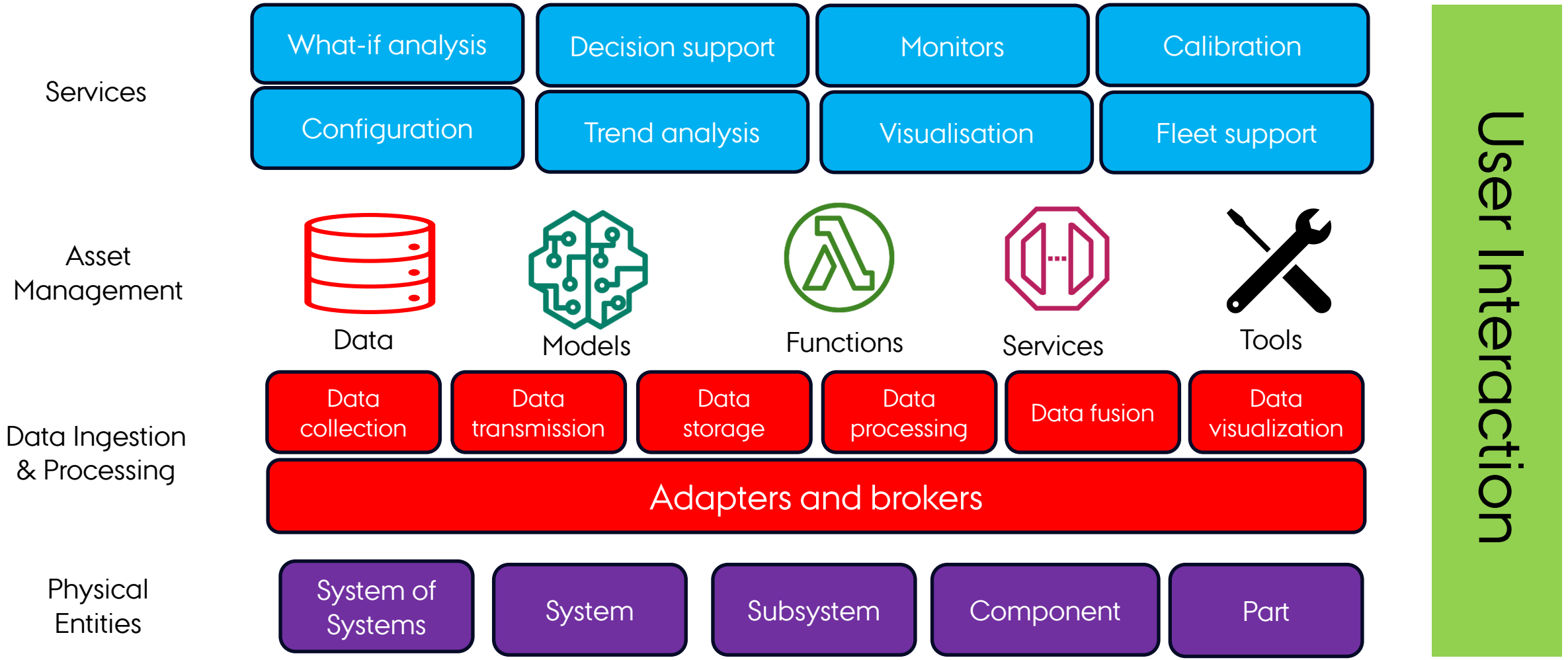
A PEEK INTO USER ROLES?



PRESENTATION OUTLINE

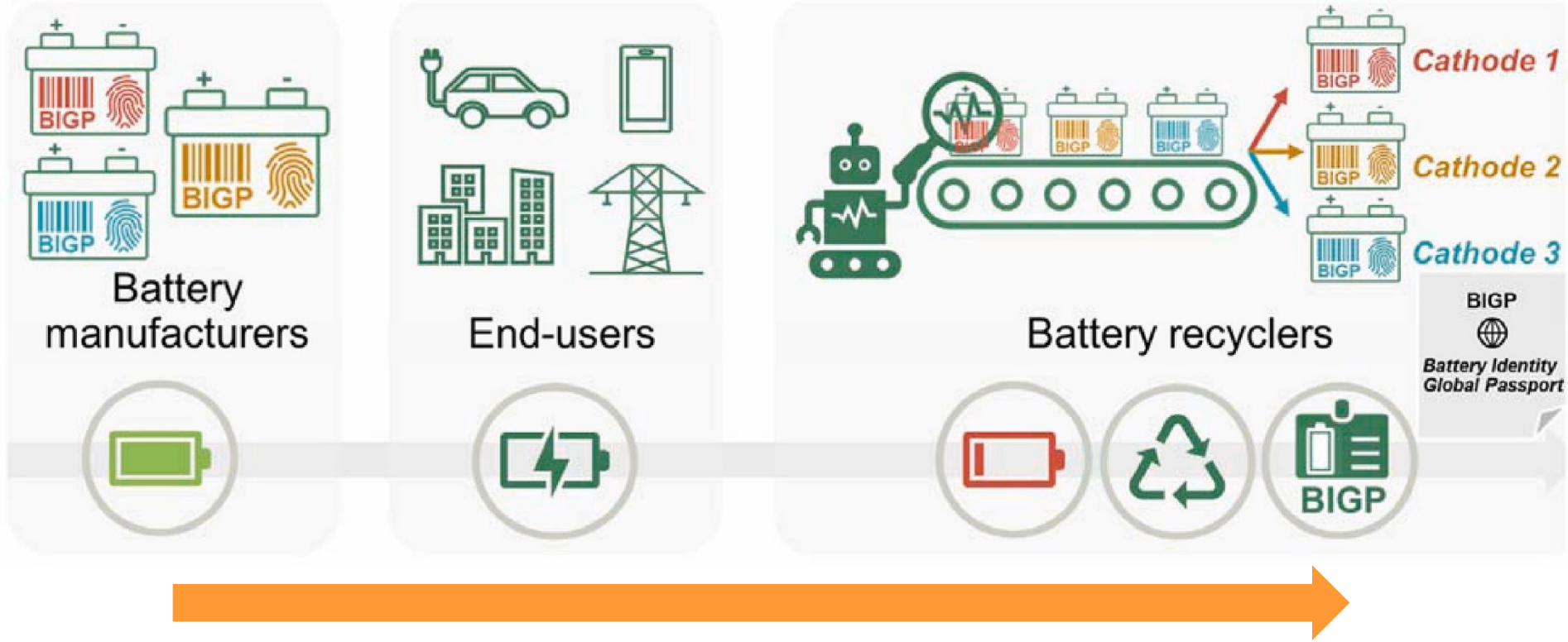
1) What is inside a Digital Twin?
2) Who are Users?
3) What are requirements for Digital Twin Platforms?
4) **One viable system architecture**
5) What is the implementation status?
6) What is to come later?
7) How can you contribute?

RECAP OF THE DIGITAL TWIN LAYERS?



NOTE: This is not a strictly layered architecture

An Example of DT/PT Lifecycle



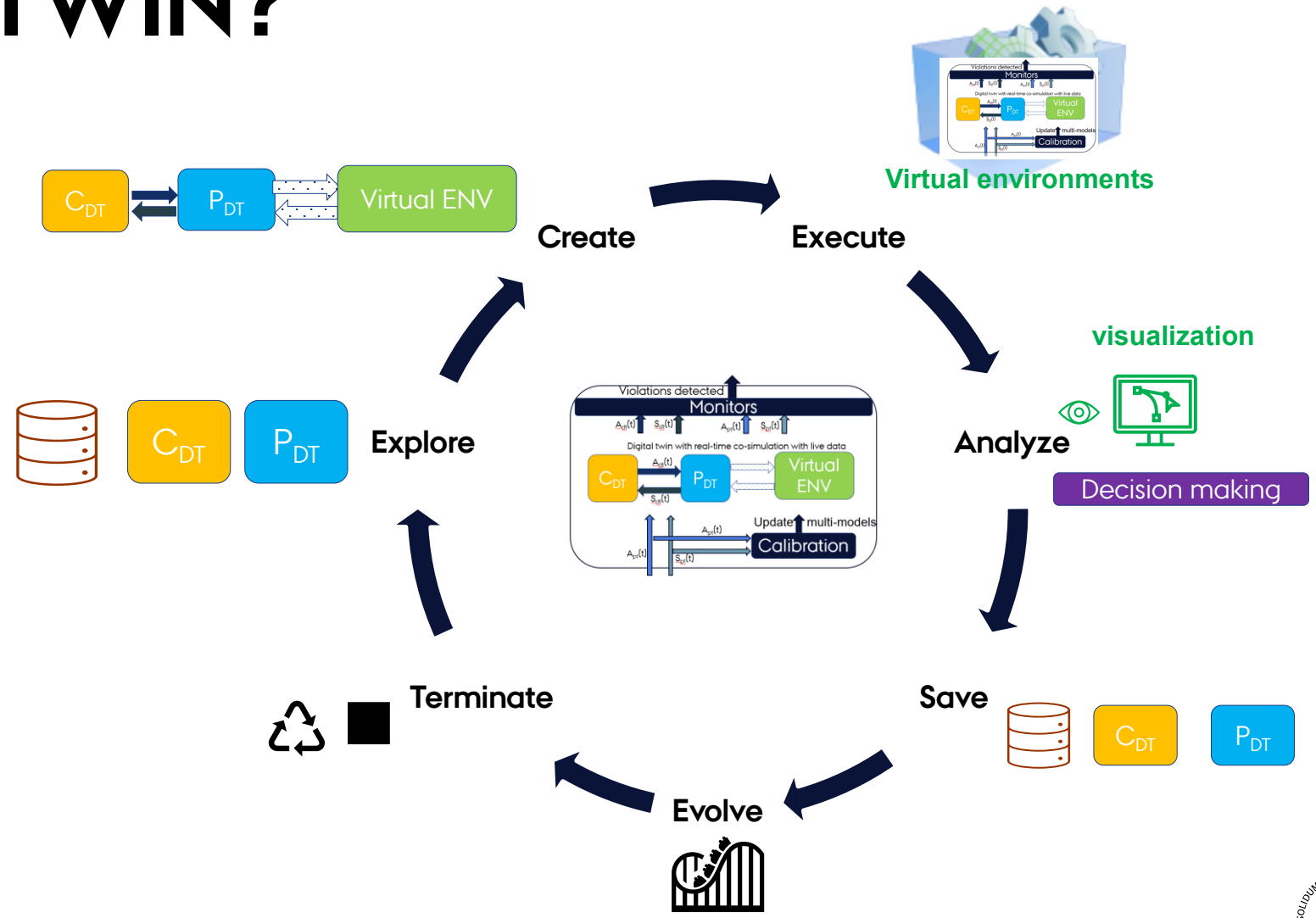
Ref: F. Naseri, S. Gil, C. Barbu, E. Cetkin, G. Yarimca, A.C. Jensen, P.G. Larsen, C. Gomes, Digital twin of electric vehicle battery systems: Comprehensive review of the use cases, requirements, and platforms, Renewable and Sustainable Energy Reviews, Volume 179, 2023,

WHAT ARE DIFFERENT LIFECYCLE PHASES OF A DIGITAL TWIN?

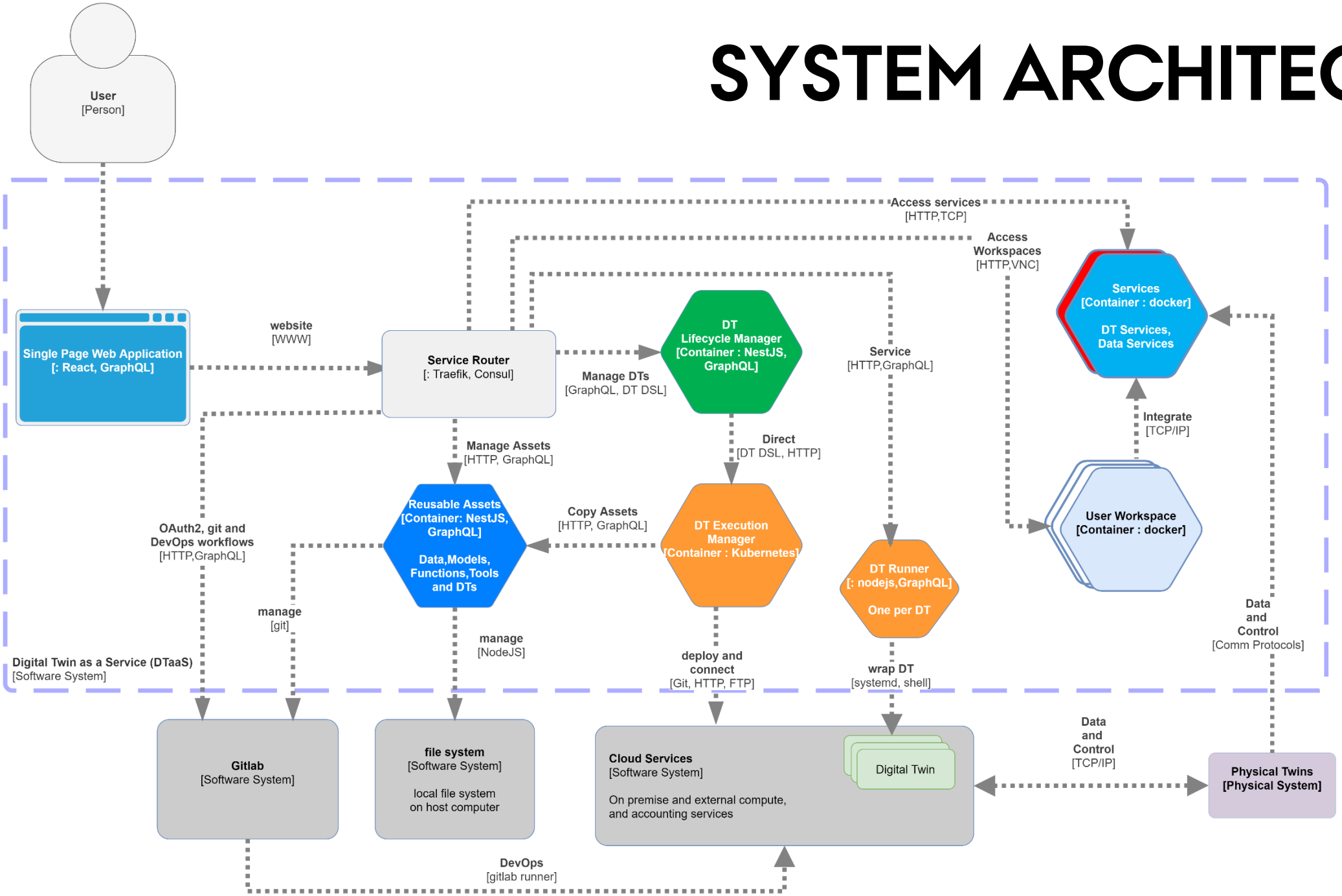
- All stages are possible
- Potentially user driven
- Transitions are not sequential

Need a dedicated

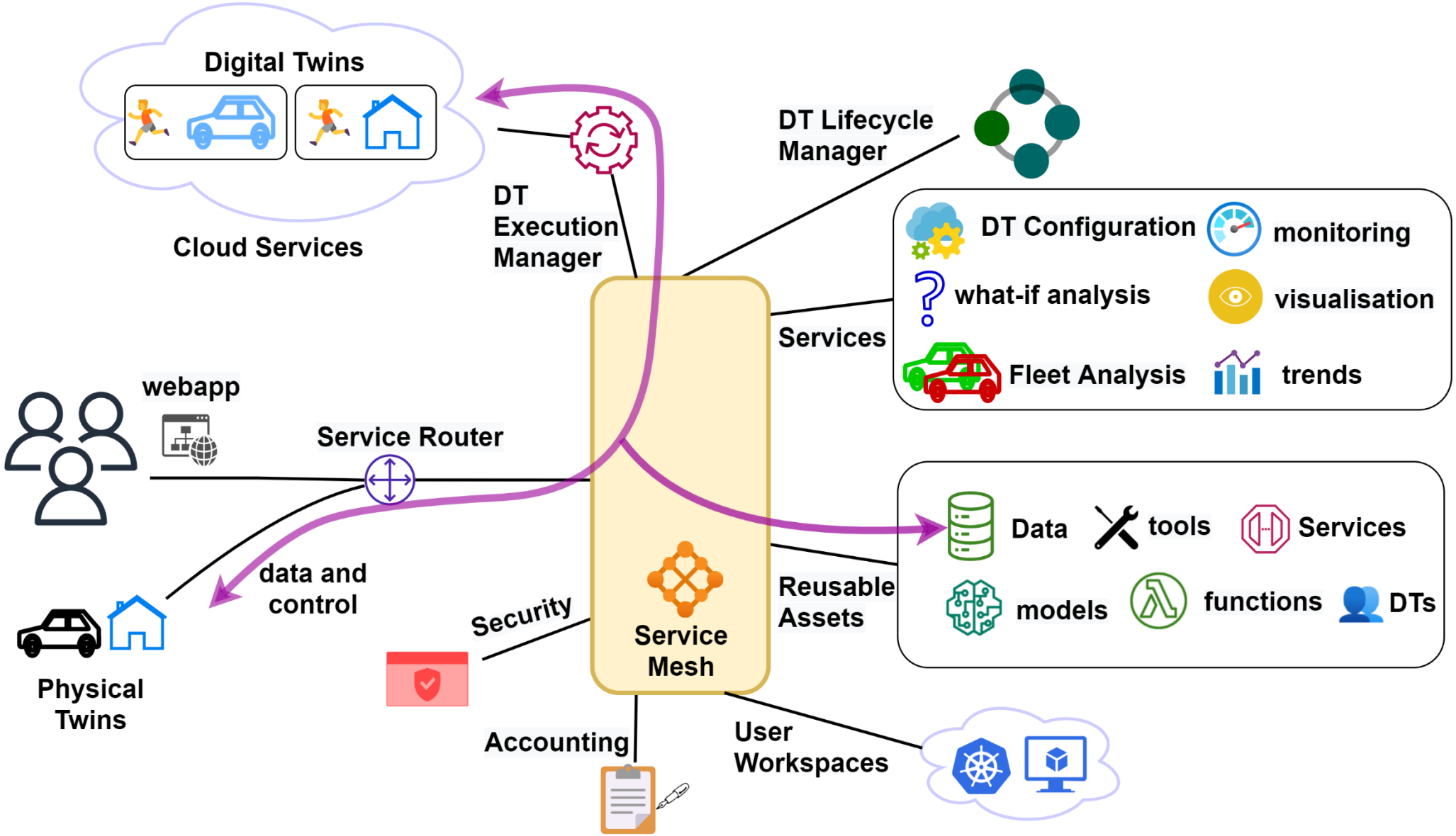
DT Lifecycle Manager
(part of Services Layer)



SYSTEM ARCHITECTURE



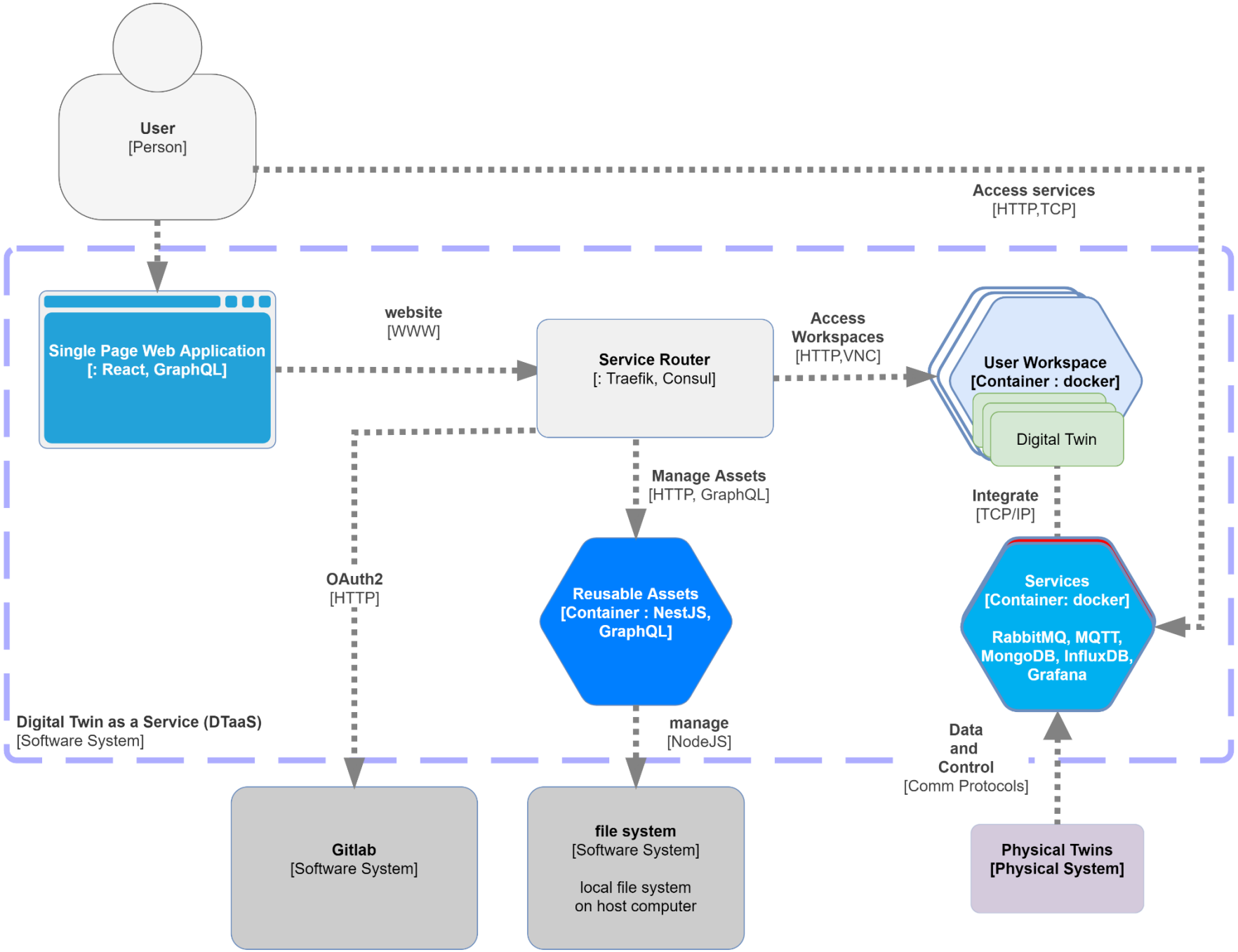
SYSTEM ARCHITECTURE: A BLOCK DIAGRAM REPRESENTATION



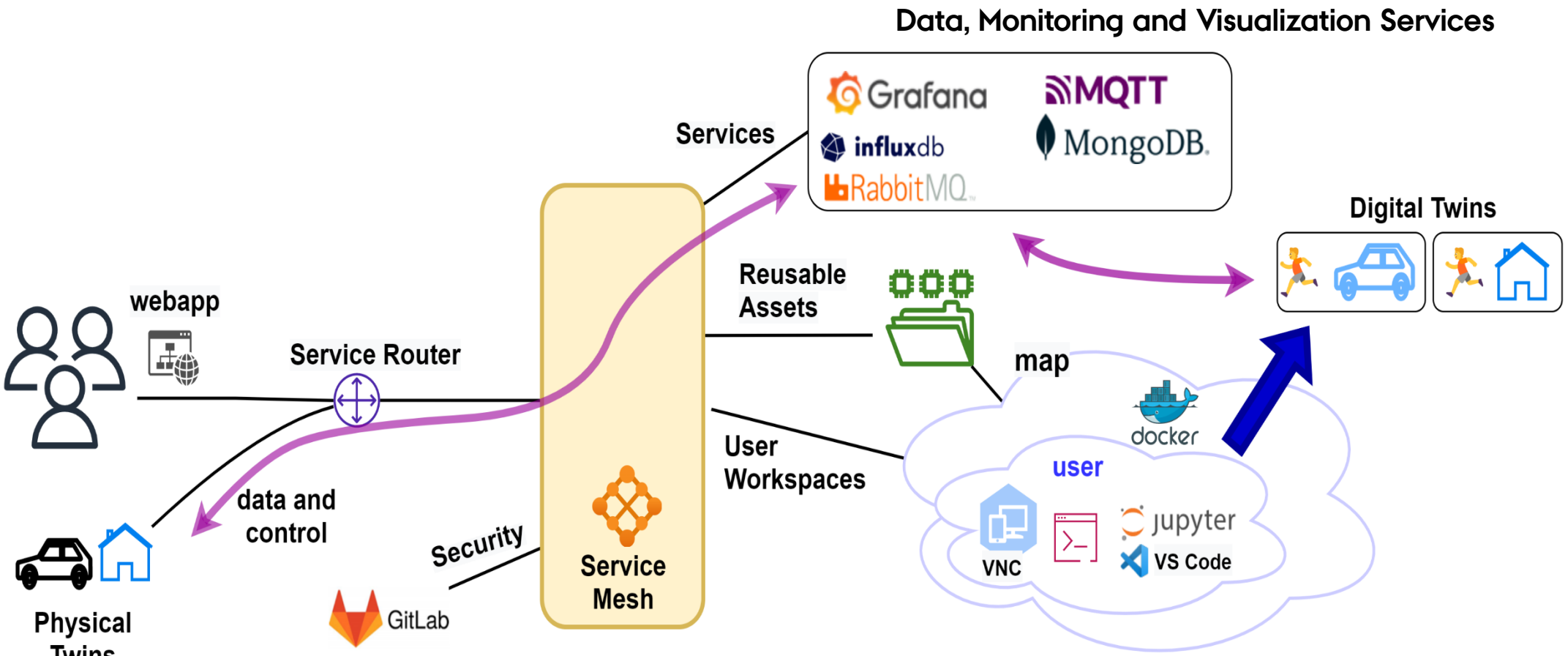
PRESENTATION OUTLINE

1) What is inside a Digital Twin?
2) Who are Users?
3) What are requirements for Digital Twin Platforms?
4) One viable system architecture
5) **What is the implementation status?**
6) What is to come later?
7) How can you contribute?

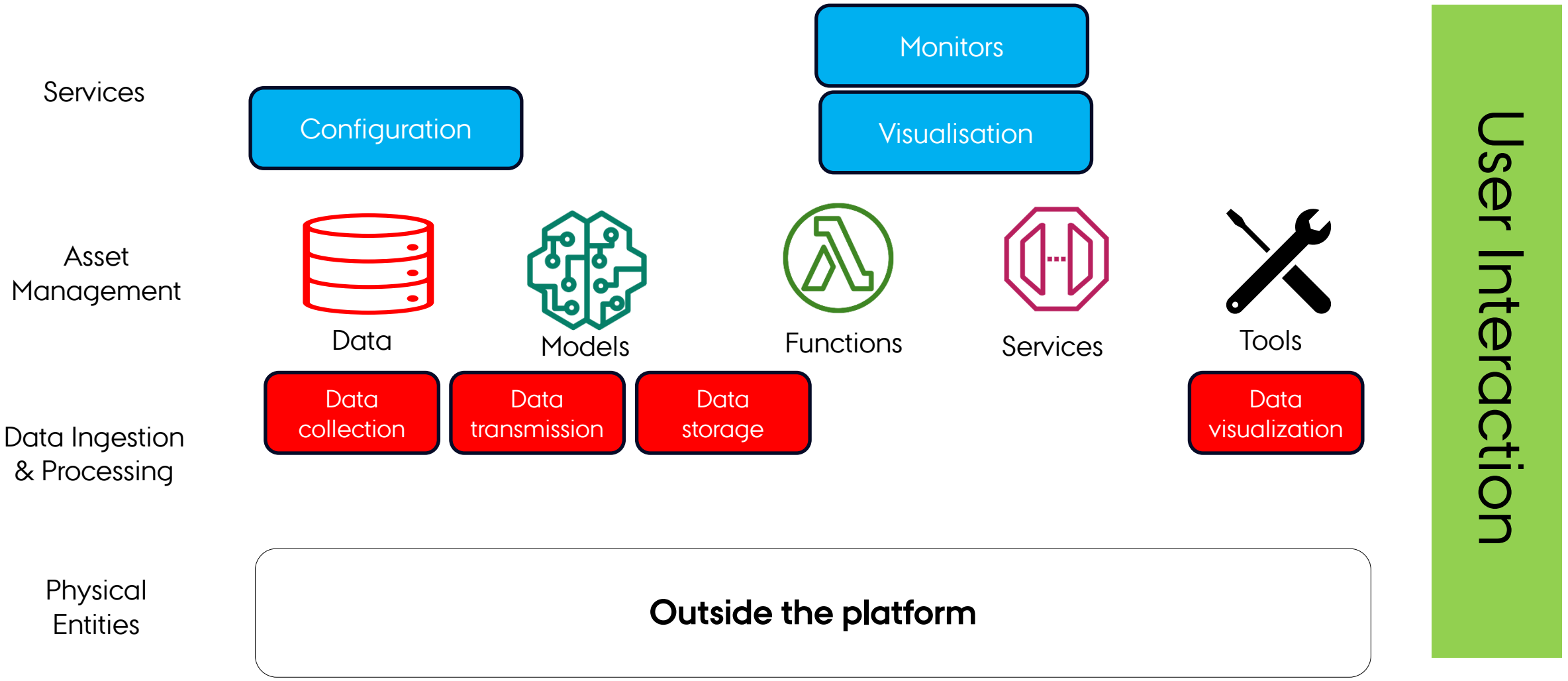
WHAT IS THE IMPLEMENTATION STATUS?



IMPLEMENTATION: A BLOCK DIAGRAM REPRESENTATION



SUPPORT FOR DIFFERENT LAYERS

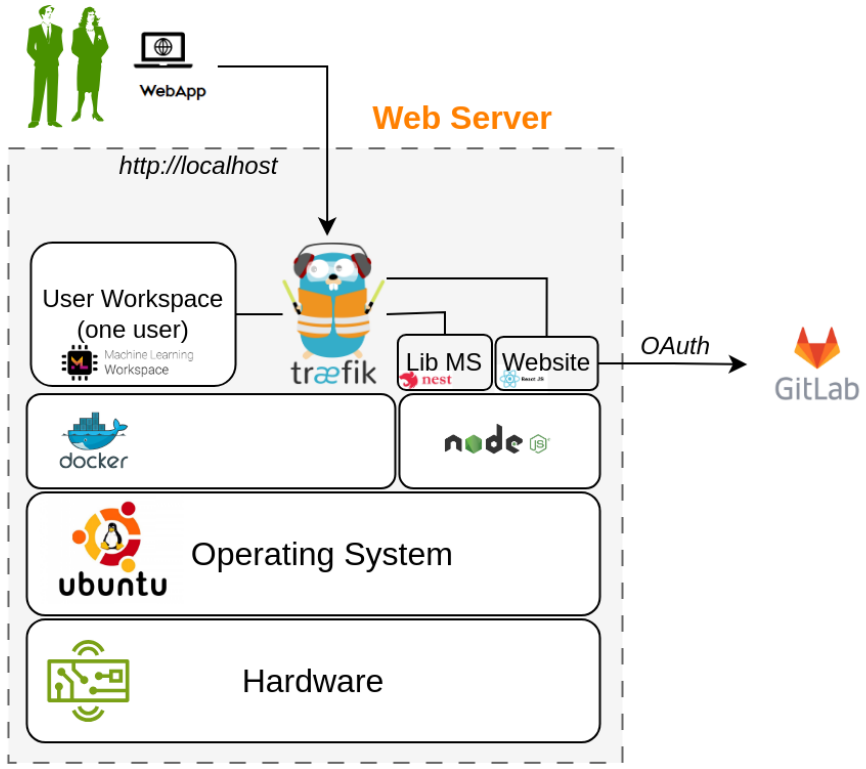


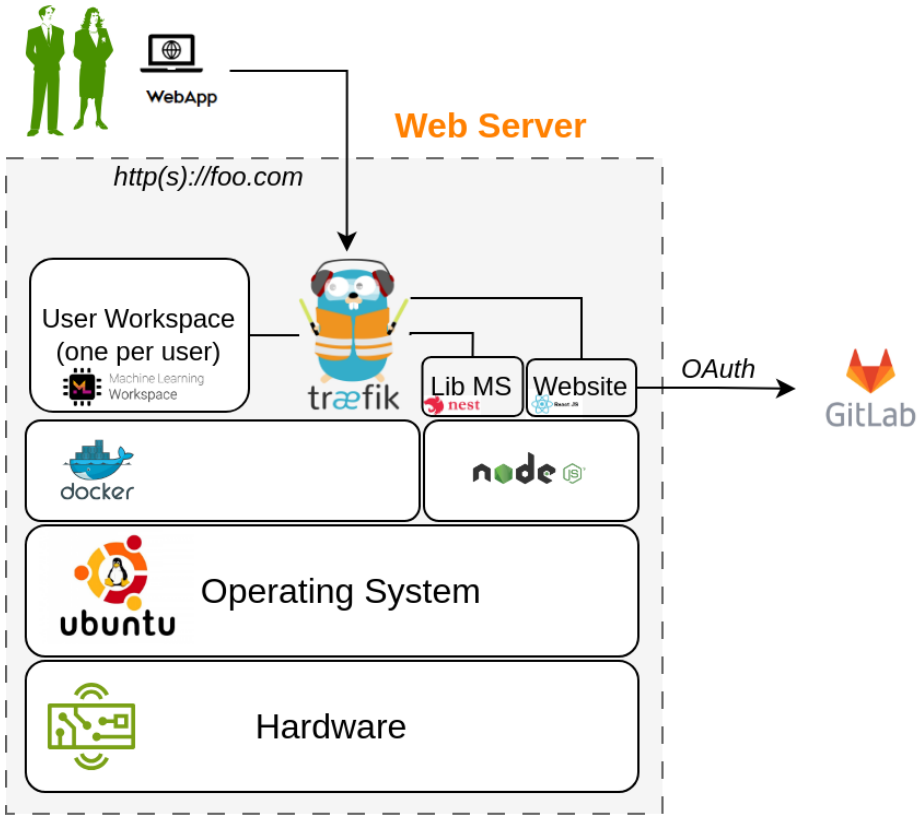
NOTE: This is not a strictly layered architecture

SUPPORT FOR DIFFERENT LAYERS (2)

Container / Component	Implementation Status	Off the Shelf Software	Temporary Replacement
Asset Management	Under Development	Octave, Matlab, File system	File system, Gitlab API
Data Ingestion and Processing	Complete	RabbitMQ, MQTT, MongoDB, InfluxDB, Grafana	
Security	Under Development	Gitlab OAuth	mTLS (additional safety)
User Workspaces	Complete (Upgrade OS and packages)	ML Workspace Docker Container	
DT Services	Not Started	InfluxDB, Grafana	
Web Application	Under Development		
Service Router	Complete (new development in service mesh)		Traefik
Execution Manager and related infrastructure	Not Started		

WHERE CAN YOU INSTALL THE SOFTWARE?





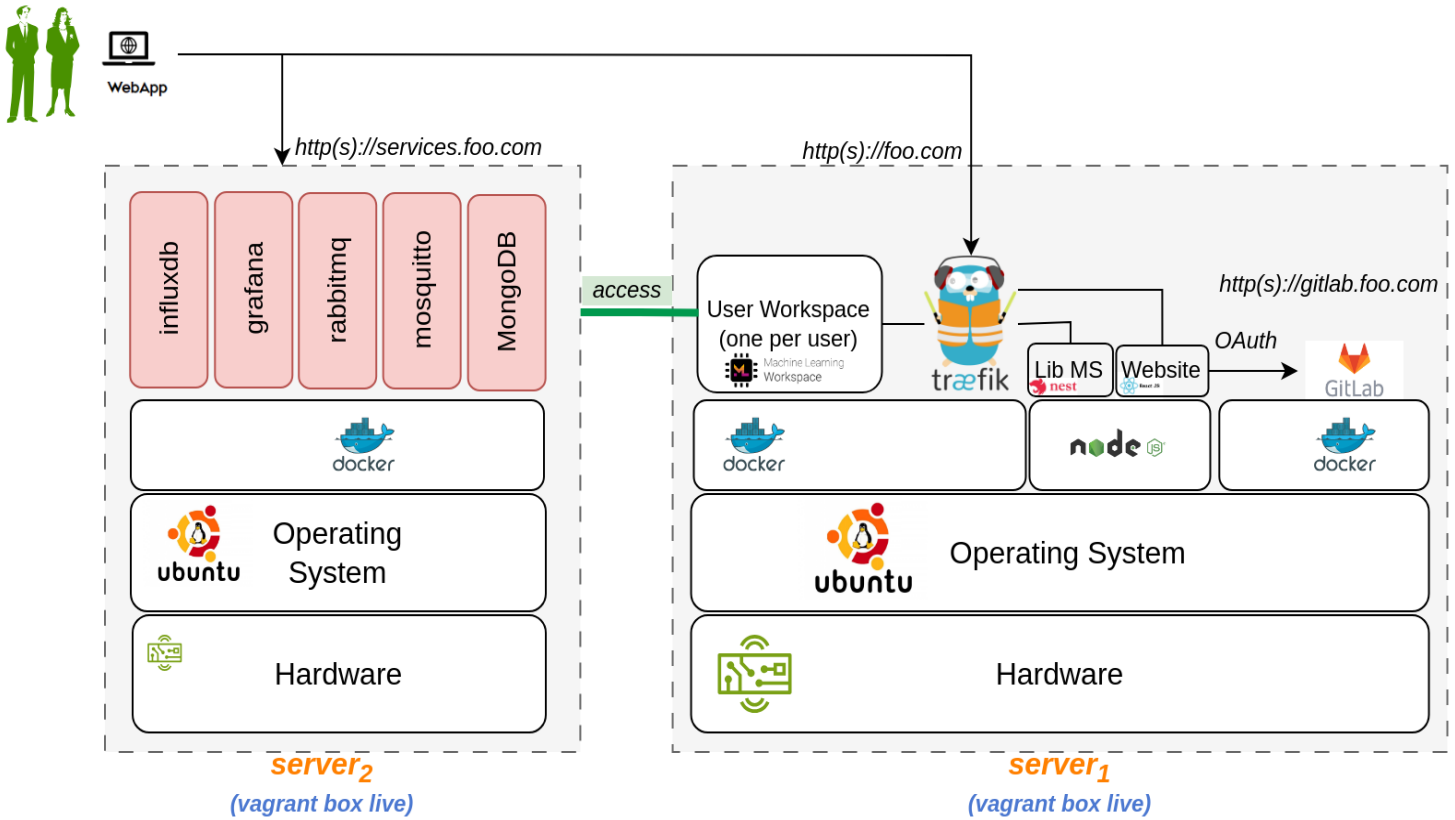
Localhost - ideal for:

- Single user
- Developement

Webserver - ideal for:

- Multiple users

INSTALL ON VIRTUAL MACHINES



Ideal for virtual machines

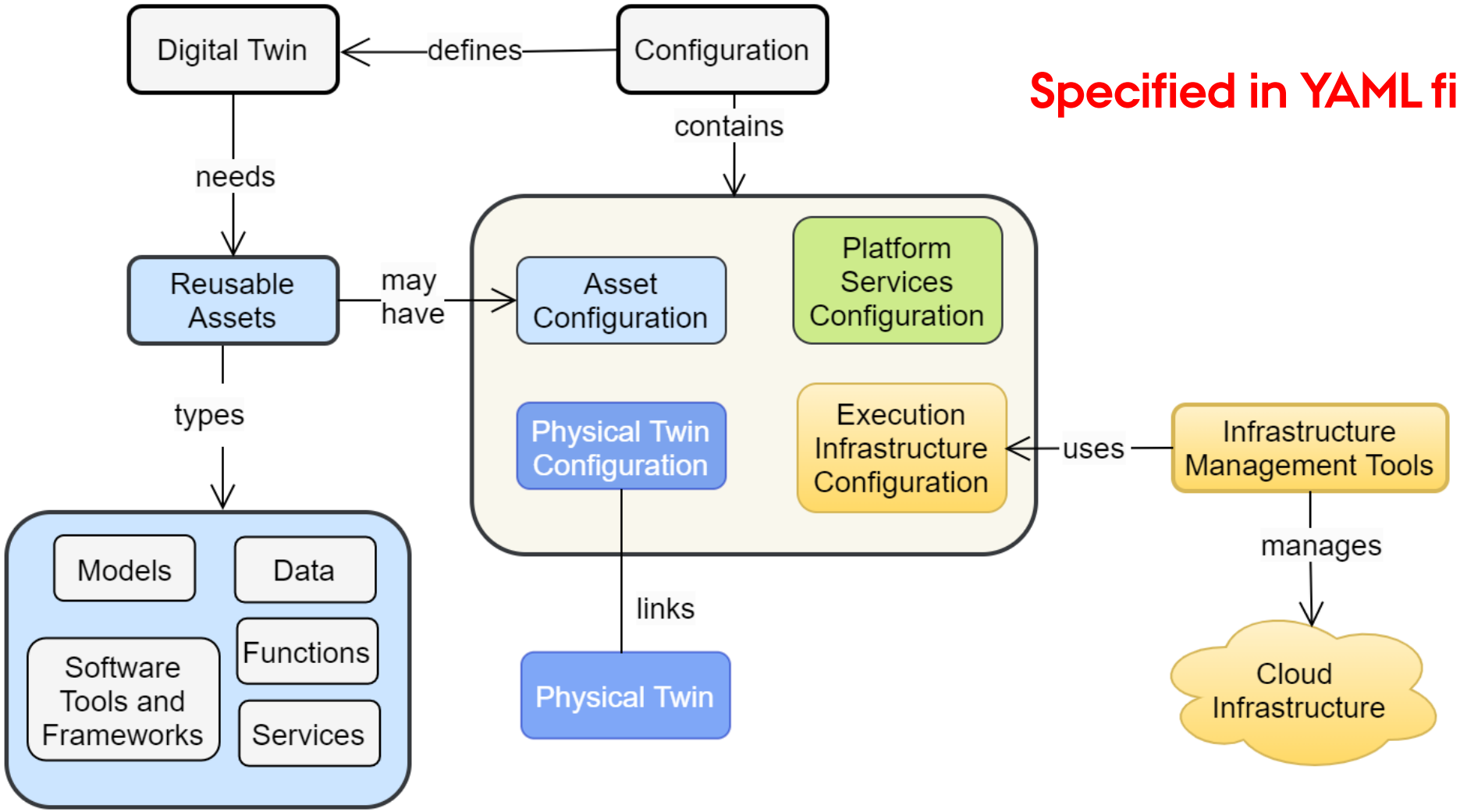
TECHNOLOGY STACK

Web Application	Asset Management Layer (Library Microservice)	Deployment/Installation Software
Typescript	Typescript	Shell
Material UI	NodeJS	Javascript
React	NestJS	Vagrant
Redux	Jest	Docker
Oauth	CloudCMD	
Jest		
Playwright		

PRESENTATION OUTLINE

1) What is inside a Digital Twin?
2) Who are Users?
3) What are requirements for Digital Twin Platforms?
4) One viable system architecture
5) What is the implementation status?
6) **What is to come later?**
7) How can you contribute?

CONCEPTUAL MODEL OF DT CONFIGURATION



Specified in YAML file

DIGITAL TWIN CONFIGURATION FORMAT (YAML)

assets: #config for all DT assets
  config:
    location: string [file, url]

compute: #execution environment
  host: enum [docker, isolated host, shared host]
  software-id: dockerid / osid

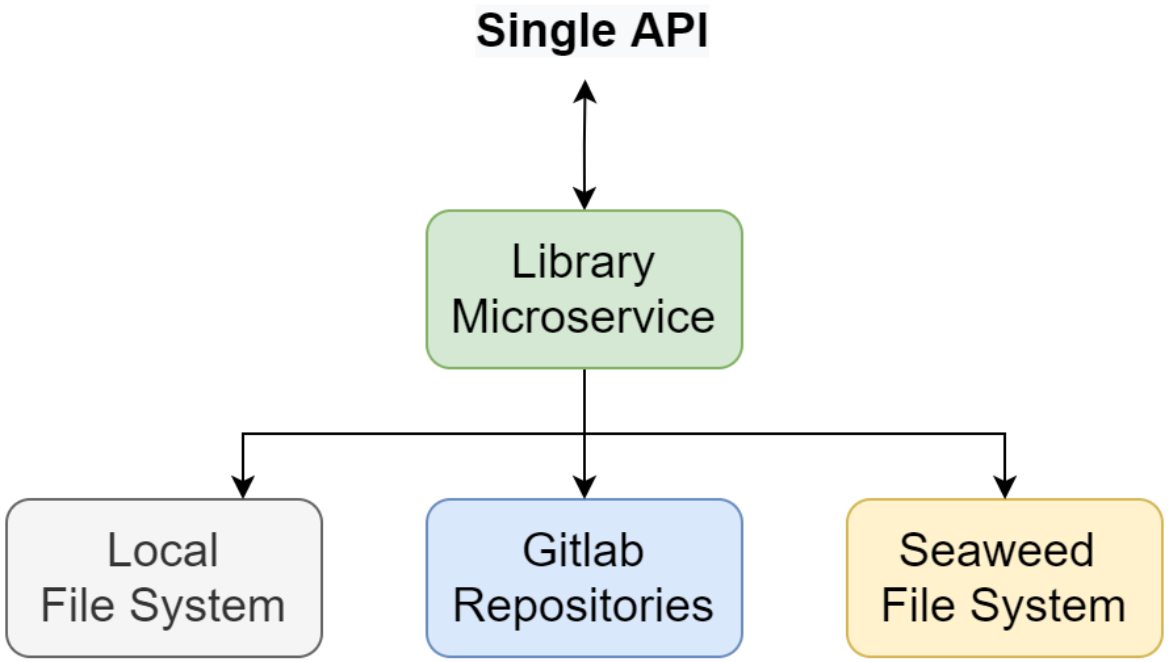
physical_twin: #info comes from PT to DT
  config:
    location: string [file, url]
    ....

services:
  internal:
    rabbitmq:
    ....

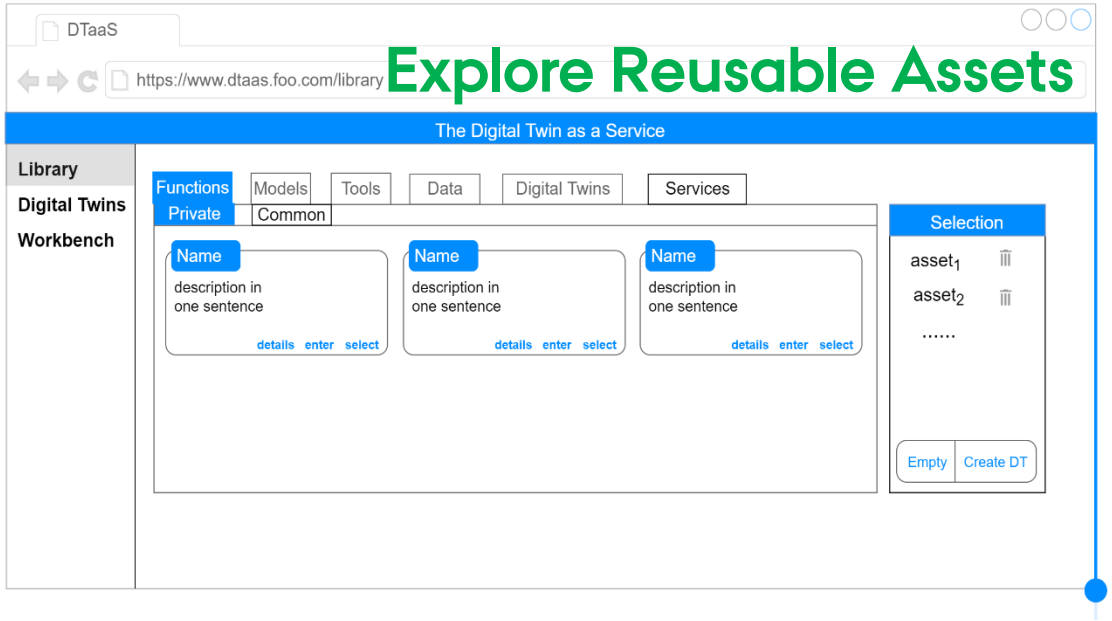
  external: #integration with external services
    service_name:
      ports:
      hostnames:
    ....

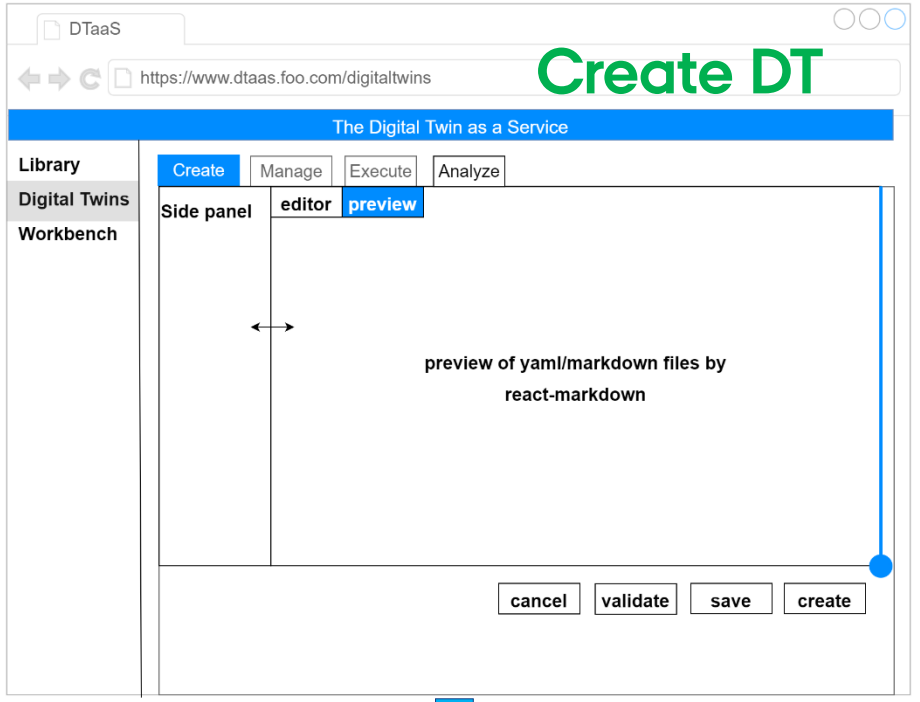
ASSET MANAGEMENT LAYER (LIBRARY MICROSERVICE)

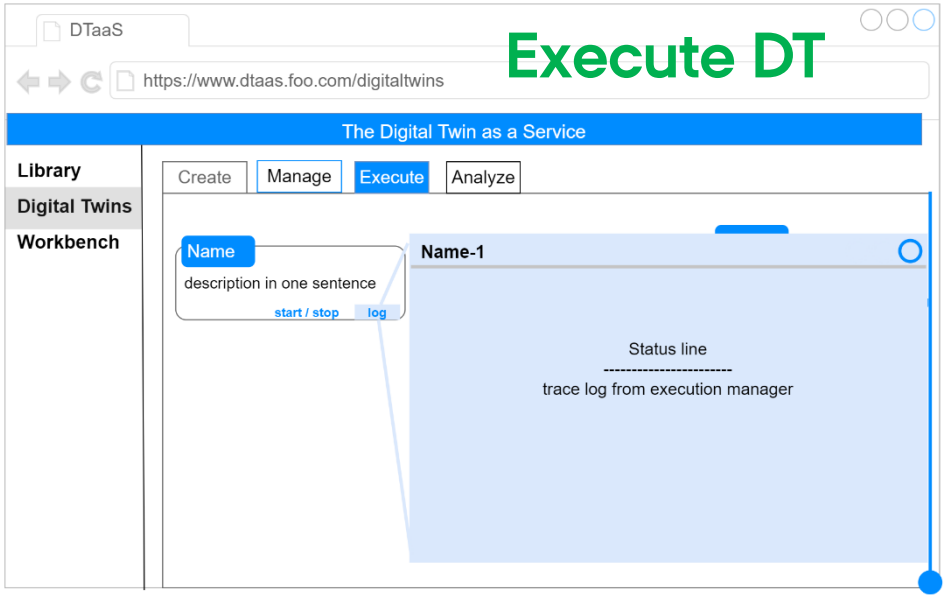
- Unified API to web application
- GraphQL API to be Gitlab GraphQL API compliant
- REST API for large file transfers

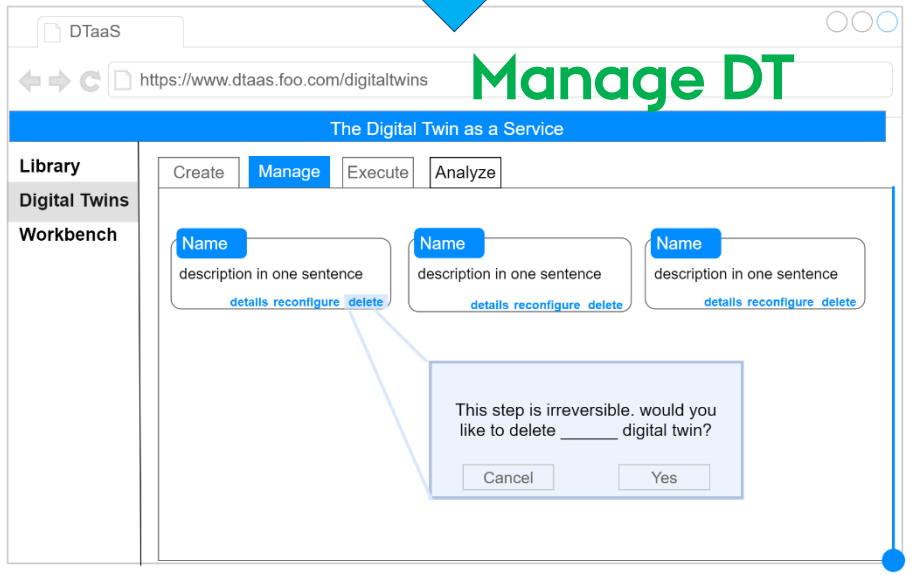


WEB APPLICATION – FLOW FOR DIGITAL TWIN DEVELOPMENT

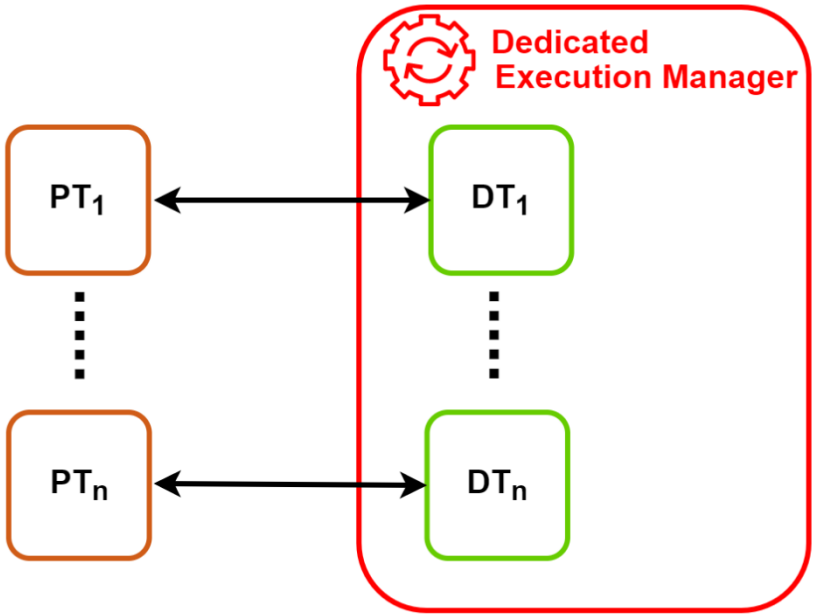






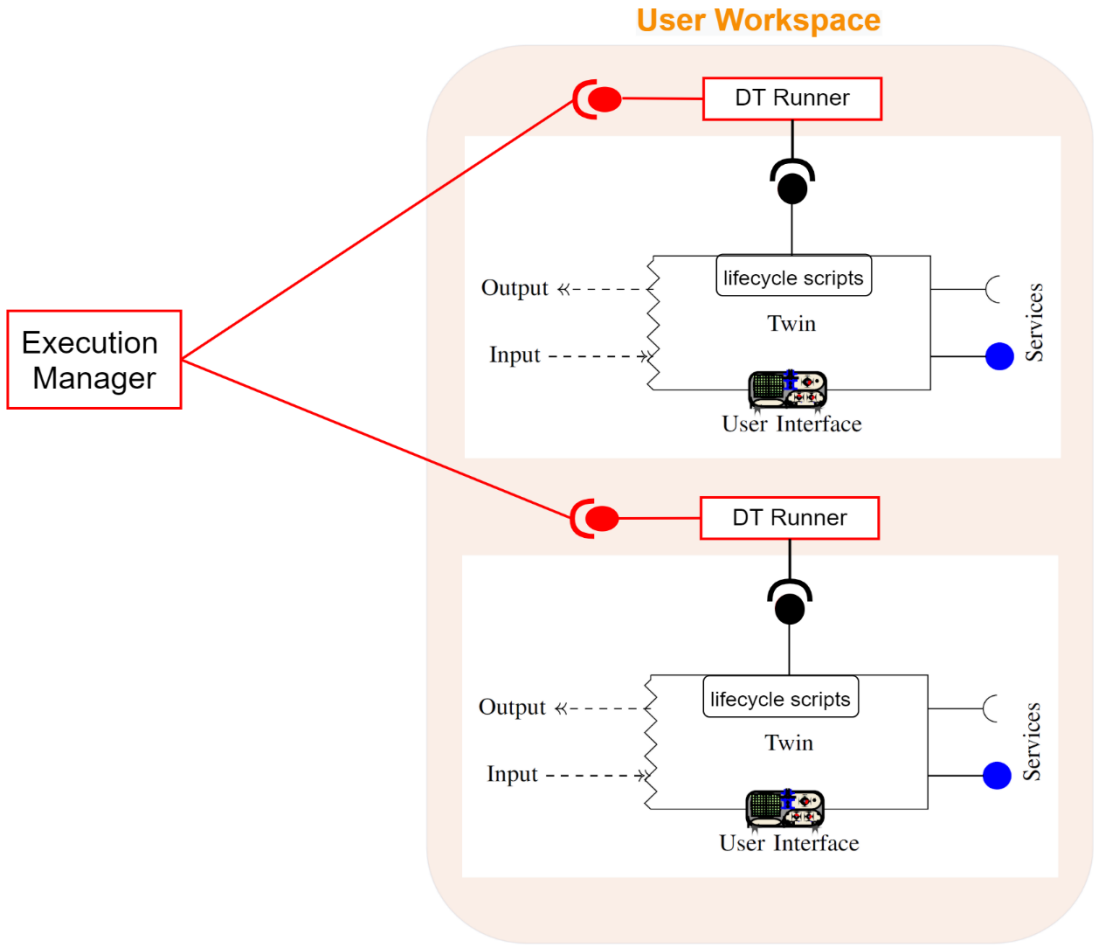


WHAT DOES EXECUTION MANAGER DO?



1. Manage execution of digital twins in cloud environments
2. Scale to many digital twins

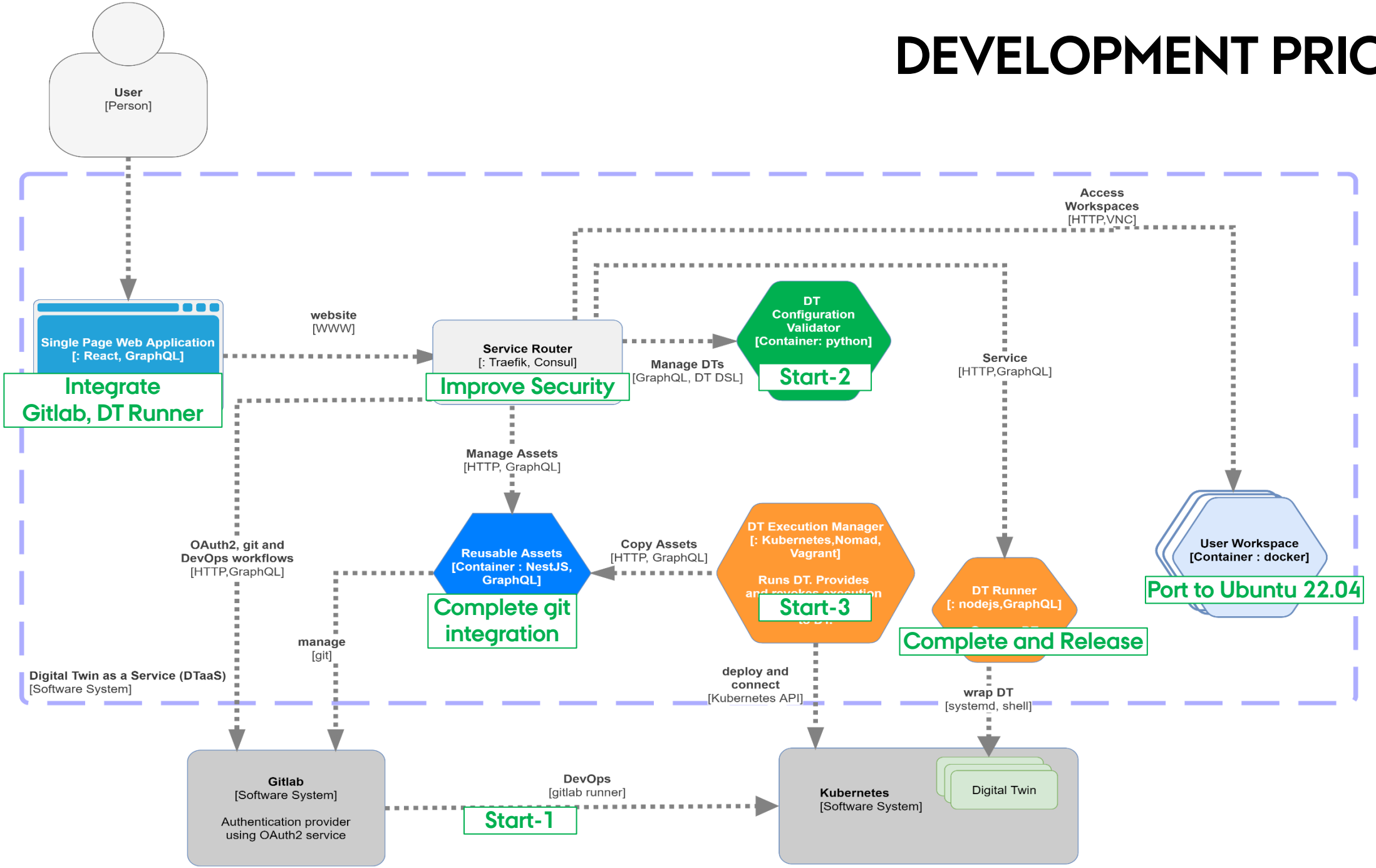
DT RUNNER



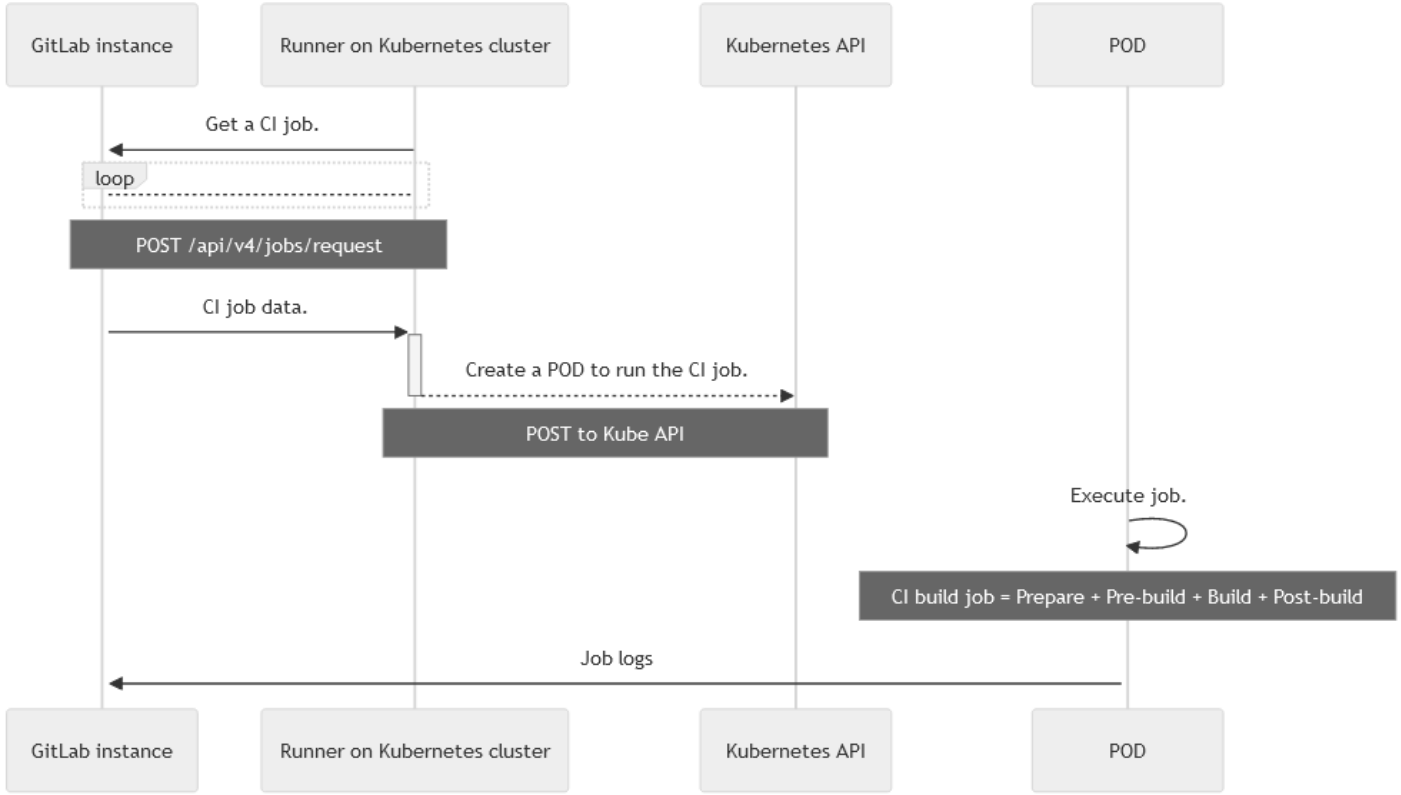
- Provides Management interface to DTaaS execution infrastructure
- Turn a DT into a web service
- Well-defined OpenAPI interface

Ref: Adopted from A Software Engineering Perspective on Digital Twin: Many Candidates, None Elected, Antoine Beugnard, IEEE International Conference on Digital Twin, Portsmouth, 2023.

DEVELOPMENT PRIORITIES



Use of Gitlab DevOps Infrastructure



Advantages:

1. Standard (Gitlab) API for all kinds of deployments
2. No need to develop own programs to manage DT deployments

Ref: Gitlab Kubernetes Executor

ON-BOARDING NEW DEVELOPERS

Your contributions are welcome

A few things to quickly get off the ground:

1. Read through the documentation to get a birds' eye view
2. Follow developer guidelines
3. Write good code
    1. follow SOLID principles
    2. Test and
    3. Maintain code quality (as measured by pre-commit hooks, codeclimate and codecov
4. Open Pull Request
5. Participate in the discussion to improve and merge the PR

RELEVANT LINKS

Use existing installation:

prasad.talasila@ece.au.dk

Research Paper

https://arxiv.org/abs/2305.07244

Docs

https://into-cps-association.github.io/DTaaS/

Codebase

https://github.com/INTO-CPS-Association/DTaaS/releases

Examples

https://github.com/INTO-CPS-Association/DTaaS-Examples