CLARIAH software & infrastructure requirements

FAIR Distribution & Deployment

CLARIAH Tech Day, 25 May 2022

Goals of the requirements

- Communicate expectations
- Document best practises and offer guidelines
- Standardize distribution & deployment for CLARIAH
- Facilitate interoperability
- Quality Control
  - Applications that are **used** must be **usable** and **reliable**.

CL	Description
A	**Excellent -** Technology adheres to as-good-as all posited infrastructure and software requirements.
B	**Good -** Technology adheres well to the requirements, there only some minor lapses
C	**Adequate -** Technology adheres to a sufficient amount of requirements, but some major ones are lacking.
D	**Lacking -** There are too many major requirements that are not met
E	**Bad -** Many requirements are not met.
F	**Unacceptable** - Technology violates or is completely dismissive of most requirements. It can not possibly be accepted without drastic changes.

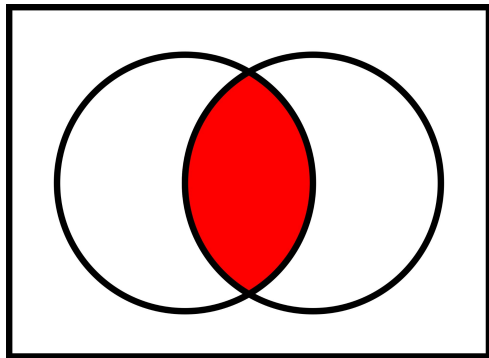# Two sets of requirements

Software requirements

For developers

How to develop, test and package
my software so it can be run anywhere

Infrastructure requirements

For operators

How to provide reliable, modern infrastructure
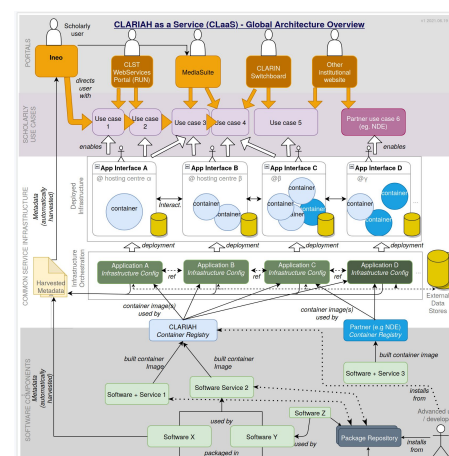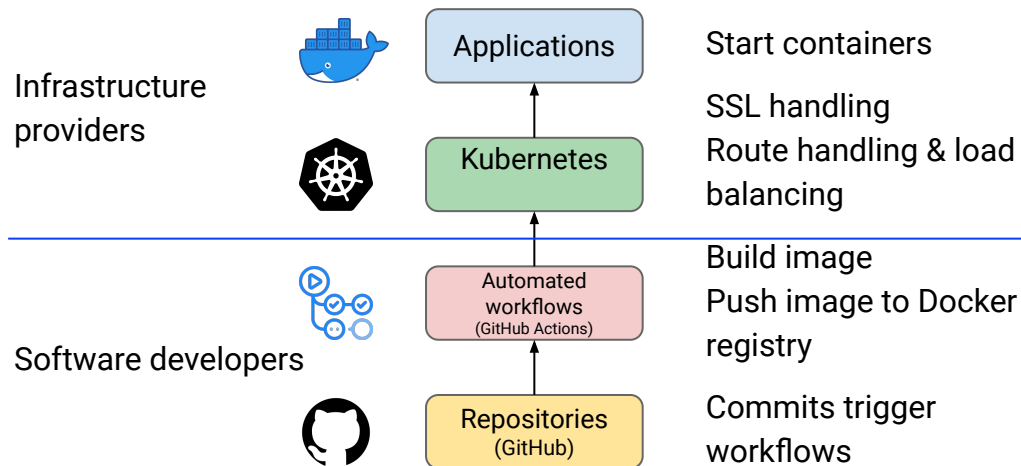so it can run any software

Requirements for Infrastructure

What is infrastructure?

Set of components that enables serving applications to users

- Runtime environment
- Build and test environment
- Code repositories
- CI/CD pipelines
- Monitoring and alerting
- Security
- Bare metal

Infrastructure providers

Software developers

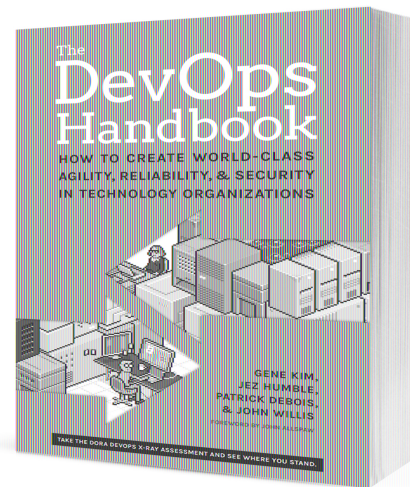|		
	Applications	Start containers
	Kubernetes	SSL handling / Route handling & load balancing
	Automated workflows (GitHub Actions)	Build image / Push image to Docker registry
	Repositories (GitHub)	Commits trigger workflows

Requirements for infrastructure

Based on experience

Best practices

Derived from work at NDE

Guiding principles

Make it reliable: fault tolerance, scaling, automated failover, change history

Automate: infrastructure as code, certificate renewal, deployment

Separate concerns: applications and infrastructure evolve independently

Decouple

Application

Packaged in containers

Configured through env vars
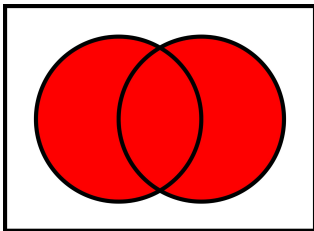
Logs to stdout

Other infrastructure? **No changes.**

Infrastructure

Run, build and test containers

Inject env vars

Capture logs

New application? **Just another container.**

Collaborate

Developers

Write application software

Package and publish software

Provide Dockerfile

Provide orchestration template

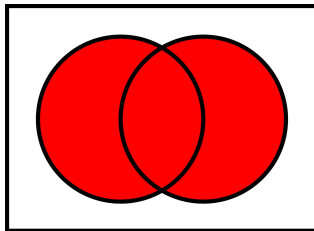Track and update dependencies

Operators

Write infrastructure

Build, run and check containers

Configure networking and storage

Manage backups

Monitor

Some requirements

Containers as interface between dev and ops

Configure applications through **env vars**

Capture logs at **stdout**

Automate builds, tests, deliveries, backups, scaling and monitoring

Manage **infrastructure as code**

Capture **metrics** and send **alerts**

Requirements for Software & Services

Software requirements following best practises

1. public version control (e.g. git) from day 1
2. a good README
3. open-source license
4. regular releases using semantic versioning
5. separation between source and configuration
6. installable using proper package managers
7. public support channel (e.g. issue tracker)
8. reusable (modular)
9. automated tests
10. software metadata at the source *(automatically harvested)*
    - *Software Metadata Requirements*
11. documentation
12. clear maintainer
13. security & privacy

https://github.com/CLARIAH/clariah-plus/blob/main/requirements/software-requirements.md

Service requirements

14. RESTful API

15. package as container

16. provide an initial template for orchestration

17. compatible with CLARIAH authentication & authorization infrastructure

18. public endpoint that provides the Web API specification

19. public endpoint that provides the metadata

Questions & Discussion

Questions & Discussion

- Do you agree with these requirements as they stand?
- Are there points in the requirements that should not be in there?
- Are you missing points in the requirements that should be incorporated?
- Are the requirements explained clearly enough?
- How are we communicating these requirements to all tool & infrastructure providers?
- Seal of approval?