

# Distribution & deployment: making dreams happen

Thomas Vermaut (WP2)

*Mario Mieldijk (lead epic D&D)*

# What is our goal? The infrastructure *just has to work!*

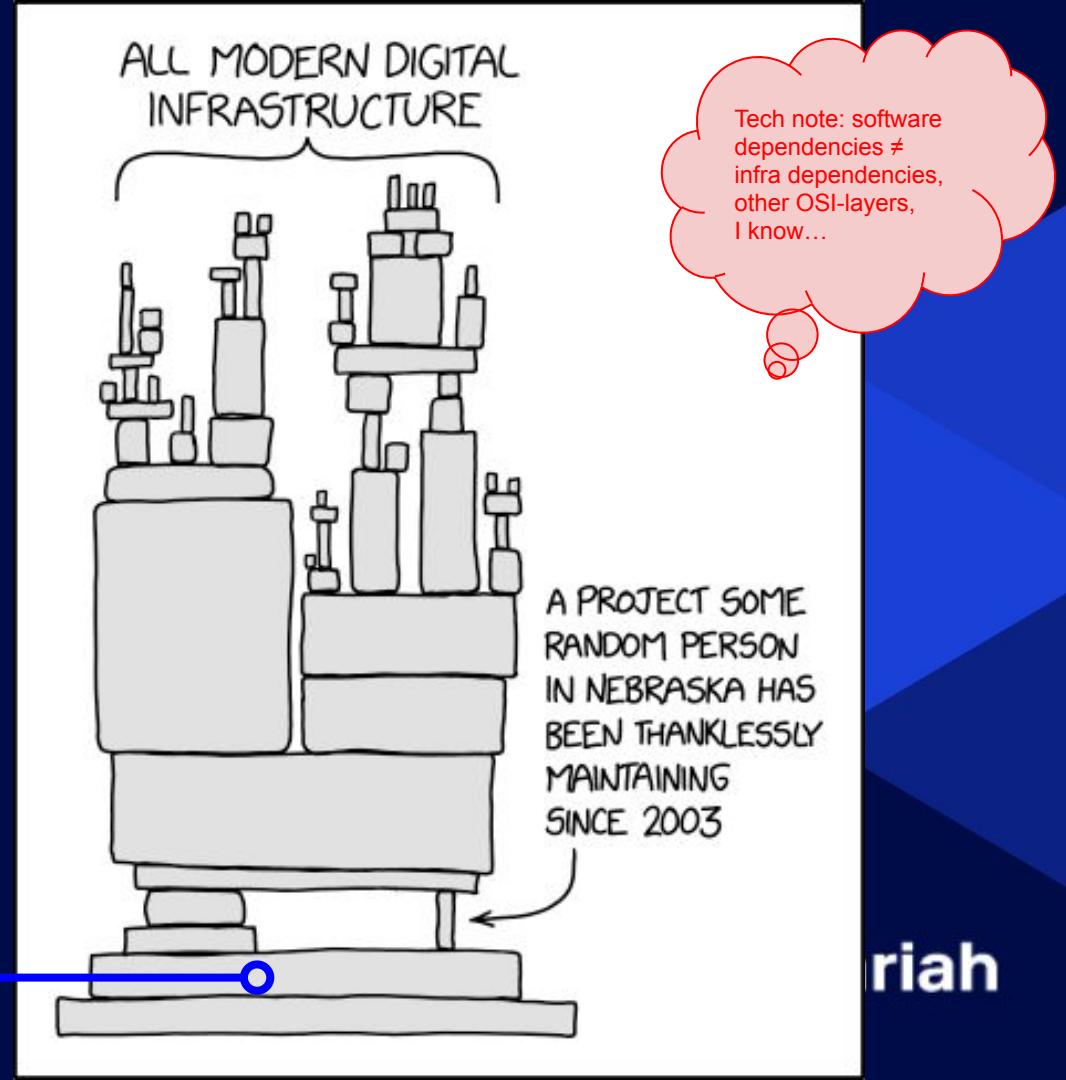
1. it has to work
  - a. it has to work WELL
  - b. it has to work, even when the context changes  
⇒ the infrastructure has to be SCalable
2. it has to be invisible
  - a. users shouldn't be hindered by useless form and formalities
  - b. cooperation across institutions should be facilitated

# Dependencies

*Someday ImageMagick  
will finally break for good  
and we'll have a  
long period of scrambling  
as we try to  
reassemble civilization  
from the rubble.*

<https://xkcd.com/2347>

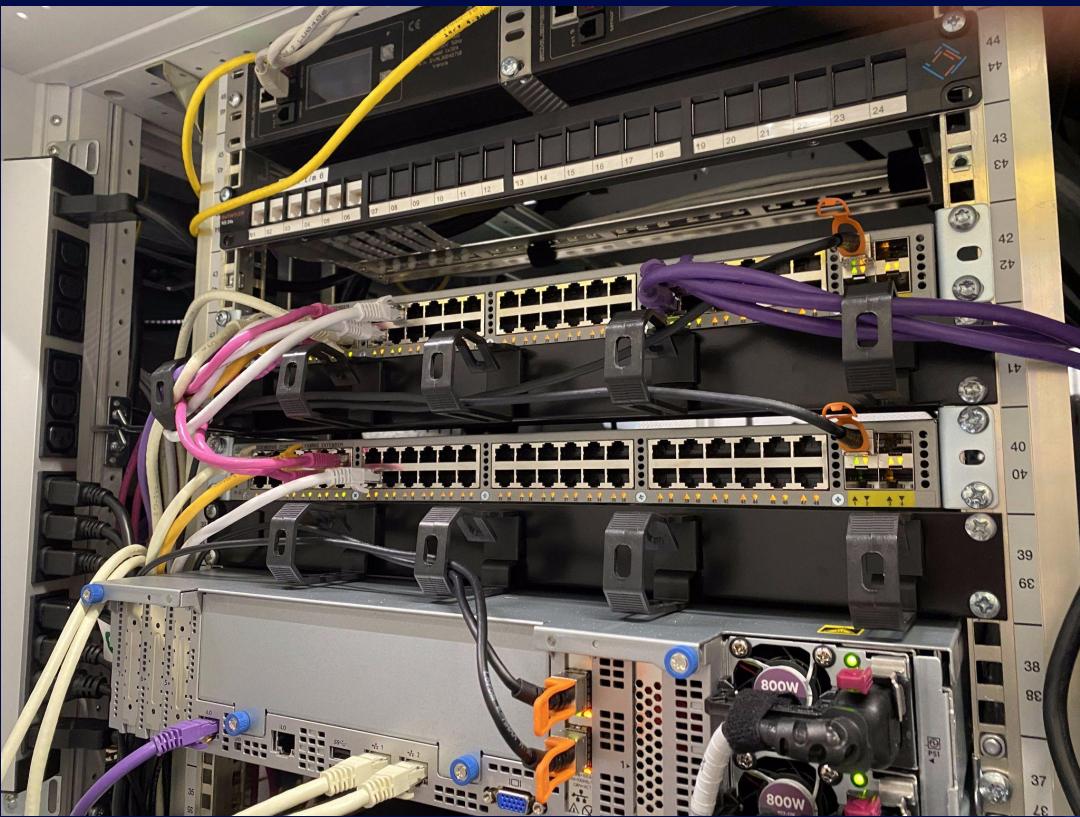
distribution & deployment?



riah

# How do we make sure the infrastructure just works? *some pics to make it tangible...*

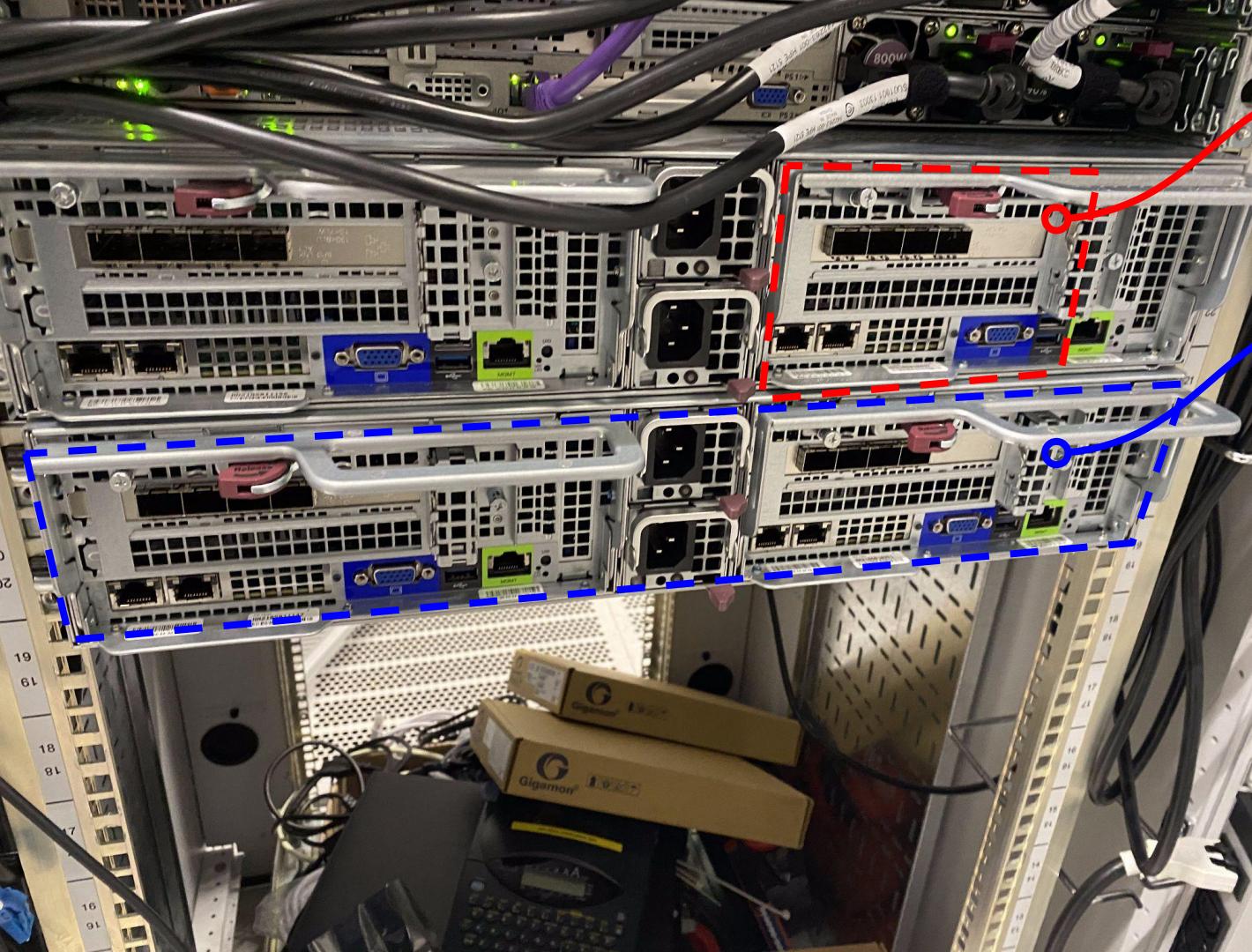




 Clariah



lariah



1 Node

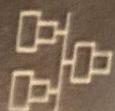
1 Block

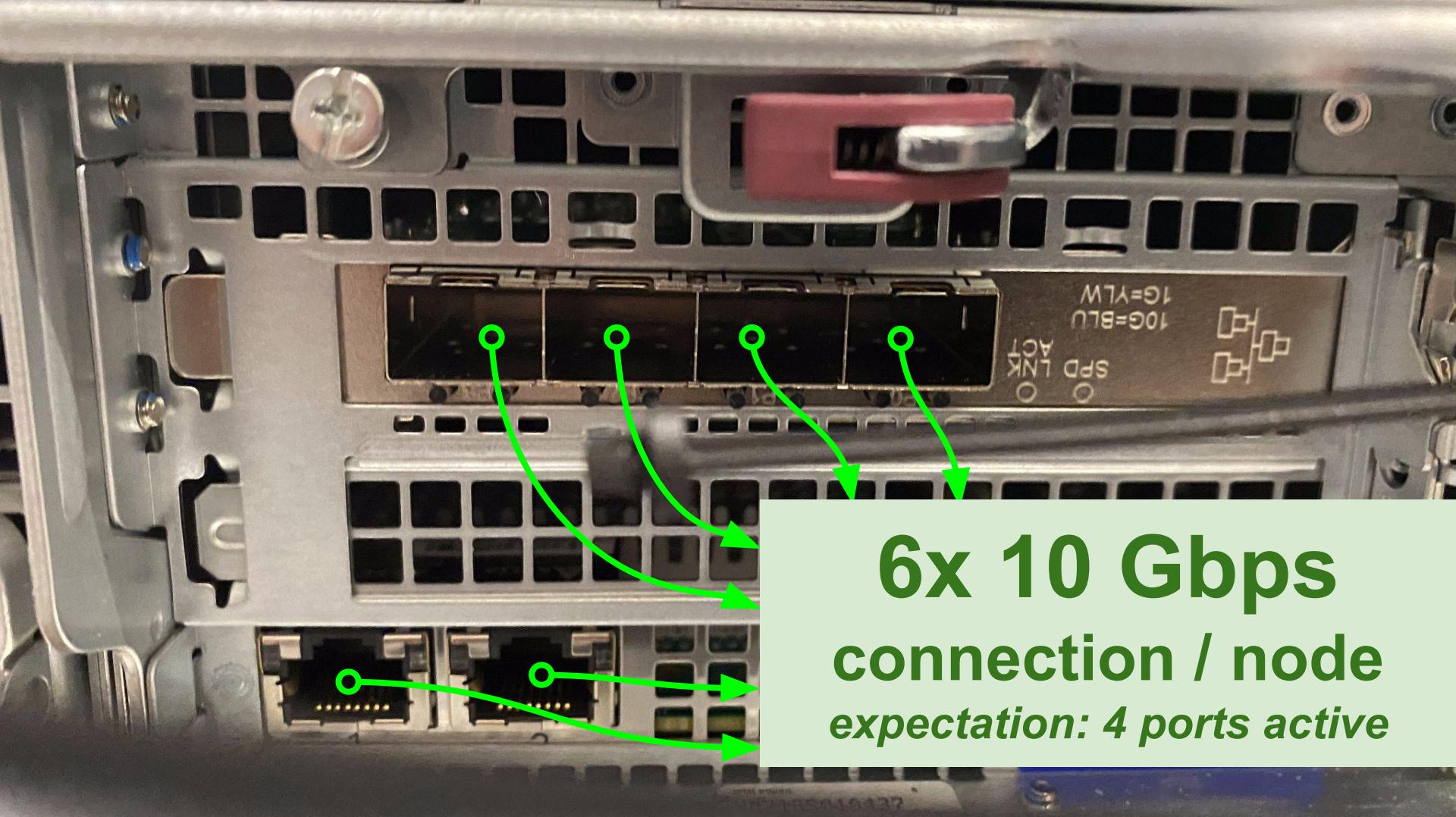
So in total:  
2 blocks  
with 2 nodes each  
= 4 nodes

1 node (failover)  
redundancy

1G=YLW  
10G=BLU

SPD LNK ACT





Topology		Information	
HBA 0: Marvell 0		Port ID	: 1
- Virtual Disks		PD ID	: 1
└ ntnx_vd		Type	: SATA SSD
└ SSD 0: Micron_5300_		Status	: Configured
└ SSD 1: Micron_5300_		Size	: 223.5GB
└ Free Physical Disks		Feature Support	: NCQ TRIM 6Gb/s 48Bits
		Current Speed	: 6Gb/s
		Model	: Micron_5300_MTFDDAV240TDS
		Serial	: 21433229DC46
		FW Version	: D3MU001
		Rotation Rate	: Solid state
		S.M.A.R.T	: OK
<b>Help</b>			
ENTER: Operation F10: Exit/Save ESC: Return			

**Hardware Summary**

4 HOSTS	2 BLOCKS	NX-8035-G7 MODEL
4 Hosts		
4 MONITORED	0 DISCOVERED	
24 Disks		
16 HDD	8 SSD	
CPU		Memory
259.2 GHz TOTAL CPU		1.47 TiB TOTAL MEMORY
Summary		
<b>STORAGE SUMMARY</b>		
Free (Physical) 271.98 TiB		
Used (Physical) 4.03 TiB (1.46%)		
Used (Logical) 2.01 TiB		
Capacity (Physical) 276.01 TiB		

# Storage: 152 TB

- 276 TiB hardware-storage
- Replication Factor 2, so: at least 138 TiB logical storage
- Nutanix Capacity Optimization  
(compression, deduplication, erasure coding)
- So: more duplication (p.e. linux-kernel-clones)  
    ⇒ more efficiency
- So: we should be able to reach 200 TB storage,  
with the same hardware.

# Infrastructure as Code (IaC) & Scalability

- Uncertainties thrive in our humanities-domain  
⇒ getting insight into usage is key!
- This is a shared problem in modern infrastructure:  
We want to be flexible, and scale in relation to the demand
  - example: you need 5 TB extra storage for your application? No problem!
  - example: you need extra CPU-power for your calculations? No problem!
  - example: Digital Humanities exploded and demand increased 10-fold?  
No problem! We can plug in a couple extra nodes,  
even without the system rebooting!
- But how do we do this kind of sorcery? With IaC of course!

# No but really, how???

Let me take you on trip...

... and let you peek into the present at our experts' desk  
and the very near future for you all !

(whatever side you will be on: provider or user)

# Step 1: user log in

or more formally  
known as:  
authentication &  
authorization

*Clariah Central  
Authentication*

The screenshot shows a web browser window with the following details:

- Page Header:** "Page 1" is displayed in the top left corner.
- Address Bar:** The URL "https://portal.dev.clariah.nl" is shown in the top center.
- Page Content:**
  - Logo:** The Clariah logo (blue circle with white 'C') is on the left.
  - Title:** "KNAW HuC Digital Infrastructure" is displayed in a header box.
  - Text:** A message instructs users to log in and select their organization from a list. It also notes that if the organization is not listed, users should consult authentication and authorization information.
  - Search Form:** A large search bar titled "Find Your Institution" with the placeholder "Your university, organization or company". Below it is a blue search button with a magnifying glass icon. Examples of what to enter are provided: "Science Institute, xxx@YourInstitution.edu, UCLA".

ah

## Step 2:

choose a  
tool you want  
to run & the  
desired  
config-option

The image shows a screenshot of a web browser window. At the top left, it says "Page 1". On the right side, there are three circular icons: two white and one blue. Below the header, there is a navigation bar with back, forward, and refresh buttons, followed by the URL "https://portal.dev.clariah.nl". The main content area has a search bar with the placeholder "Search for a tool" and a magnifying glass icon. Below the search bar, the word "Results" is displayed. A list of four options is shown in a box, each preceded by a checkbox:

- Option 1
- Option 2
- Option 3
- Option 4

At the bottom right of the content area is a blue button with the word "Ok".

# Step 3: system creates option 3

The System creates the environment you need this will take 5 minutes

Would you like to receive an email when the environment is ready?

Yes      No

Page 1

https://portal.dev.clariah.nl

# Step 3b: system generates config:

Option 3 Technical metadata

4 Cpu

4 GB ram

20GB disk

Unique URL ree-sucher.option3.dev.clariah.nl



```
playbook.yml
---
- name: Provision VM
  hosts: localhost
  gather_facts: false
  vars:
    api_version: "3.1"
    cluster_url: nutella-00.diginfra.net
    prism_user: superadmin
    prism_password: "SuperComplexPasswordYouWillNeverGuess!"
    cluster_name: "nutella-00"
    subnet_name: "vl-356"
    image_name: "focal-server-cloudimg-amd64"
    my_vms:
      vm_name: ree-sucher.option3.dev.clariah.nl
      ip: '10.27.4.192'
      dnsname: "{{ vm_name }}"
      ram_mb: 4096
      num_cpu: 4
      disk1:
        size_gb: 20
      customization_option: "3"
  tasks:
    - name: Provision VM
      include_role:
        name: clariah_vm_provisioner
  vars:
    vm_defs: "{{ my_vms }}"
    send_notification_mail: yes
~
```





nutella-00

VM



Overview · Table

VM

Recent Tasks

VM create

1%

✓ Received operation to create VM re-  
sourcer.option3.dev.clariah.nl.

 Incl

VM create

1

[View All Tasks >](#)

VM Name

Host

IP Addresses

Cores

Memory Capacity

Storage

CPU Usage

Memory  
Usage

Controller Read IOPS

Summary

## VM SUMMARY

Total VMs

4

- Powered On 4
- Powered Off 0
- Suspended 0

Total Provisioned vCPU

48

## Performance Summary

## Cluster-wide CPU Usage



## Cluster-wide Memory Usage

ariah

```
Ubuntu 20.04.4 LTS ubuntu tty1
```

```
ubuntu login: _
```



<https://ree-surcher.option3.dev.clariah.nl>

Welcome page Option 3

From

Subject

To

CC

BCC

Hi,

Thank you for using our services.

Your new environment can be reached on

<https://ree-surcher.option3.dev.clariah.nl>

Good luck with your research

Clariah Support team

contact: <https://support.dev.clariah.nl>

# Authentication & Authorisation

~ Satosa

Are you really who you say you are?

Can you do what you are trying to do?  
i.e. are you allowed to do so?

Why are we making such a big deal about this?  
Is it not plain and simple?

```
if (password == 'P@$$w0rd!') {  
    do Ineo();  
}  
else {  
    panic;  
    try again;  
}
```

# No!



# With that out of the way...

- Authentication & authorisation for **software**:
  - simple: using authentication-service!
- A&A for **services / CMD-usage**:
  - a bit more tricky with a token / key, but very doable
- Doing **Tools2Data** with A&A?
- Crossing institutional borders?
- Deriving authorisation from authentication?



# What's the big thing?

- How do we know what a user is authorised to do?
- We can't keep track all of possible combinations of Users, Software-services & Data  
Or rather: enumerating those would be horrible...  
...and keeping track of all users, cross-institutions a nightmare!
- We need to be able to link a key to a usage-role,  
so a user using the key for a certain role,  
can be given the authorisations of that role.
- This has consequences for how institutions link up with Clariah-authentication  
and thus requires some talking to each other, and agreement on this issue.

Hence the importance of the CLARIAH Tech Days  
& our tech Requirements!



# Software & Infrastructure Requirements

## Governing principles:

- SEP: separation between code (the application) and the infrastructure it runs on
- AUTO: automation to reduce human error and increase transparency
- REL: reliability

## Requirements: descriptive vs prescriptive

# Infrastructure Requirements:

<https://github.com/CLARIAH/clariah-plus/blob/main/requirements/infrastructure-requirements.md>

1. The infrastructure MUST run applications that are packaged as OCI containers. (SEP)
2. The infrastructure MUST connect to a container registry. (SEP)
3. The infrastructure MUST be able to run stateful applications.
4. The infrastructure MUST configure applications through environment variables. (SEP)
5. The infrastructure MUST capture application logs at stdout. (SEP)
6. The infrastructure MUST aggregate logs. (REL)
7. The infrastructure MUST automatically build the application. (AUTO)
8. The infrastructure MUST automatically run application tests when commits are pushed to the application repository. (AUTO)
9. The infrastructure MUST automatically deliver new application versions to an acceptance and/or production environment. (AUTO)
10. The infrastructure MUST be highly available corresponding to the infrastructure supplier's SLA (REL)
11. The infrastructure MUST expose applications at public HTTPS web endpoints.
12. The infrastructure MUST provision and renew TLS certificates for the web endpoints. (REL)
13. The infrastructure MUST back up all application data from deployed services at an agreed upon interval. (REL)
14. The infrastructure MUST be GDPR-compliant, for instance in the way it stores data.
15. The infrastructure MUST be configured with security in mind. (REL)
16. The infrastructure MUST store, manage and deliver secrets and other sensitive data to application (REL)
17. The infrastructure SHOULD support zero-downtime deployments. (REL)
18. The infrastructure configuration SHOULD be declared in code. (AUTO)
19. The infrastructure SHOULD capture metrics. (Should have, REL)
20. The infrastructure SHOULD send alerts. (AUTO)
21. The application repository SHOULD be open source. (REL)
22. The infrastructure SHOULD automatically configure DNS. (AUTO)
23. The infrastructure SHOULD automatically scale when usage changes. (AUTO)
24. The infrastructure MAY have an optimized way of hosting static files. (REL)

# Software Requirements:

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

1. The software's source code **MUST** be stored in a public version control system (VCS).
2. A README file **MUST** be provided in the root directory of the VCS
3. The software **MUST** be distributed as open source under an OSI approved license
4. The software **MUST** be released periodically with clear version numbers
5. The software **MUST** separate code from configuration.
6. Each release of the software **SHOULD** be installable through a proper package manager
7. The software **SHOULD** have a public support channel
8. Software **SHOULD** be reusable
9. Software **SHOULD** come with automated tests
10. Software **MUST** define software metadata along with the source code
11. Software **SHOULD** be documented.
12. Software **MUST** have a clear maintainer
13. Software **MUST** be developed with attention to security & privacy
14. Services **SHOULD** provide a simple RESTful API
15. Services **MUST** be packaged as containers
16. Service developers **SHOULD** provide an initial template when multi-container orchestration is needed
17. Services **MUST** be compatible with CLARIAH's authentication and authorization infrastructure
18. Services **MUST** expose a public endpoint providing their specification
19. Services **SHOULD** expose a public endpoint providing metadata

# Questions?