university of groningen / center for information technology

High Performance Computing

From Complexity to Simplicity

Our Cloud Infrastructure Redesign

David Visscher
Max Roeleveld
2025-05-22

Who are these people!?

David Visscher

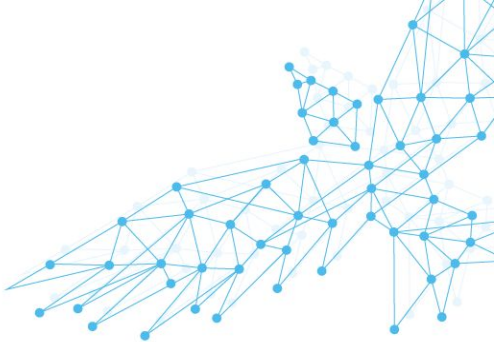
Architect of the Cloud4 project at the University of Groningen.

Max Roeleveld

Cloud engineer in the High Performance Computing team at the University of Groningen.

university of groningen / center for information technology

What do we do?



- Team HPC: **High Performance Computing**
- Provide computing services for science and education
- Built, run, and operated by us, at the University of Groningen

university of groningen / center for information technology

Space Exploration

Euclid is designed to explore the evolution of the dark Universe. It will make a 3D-map of the Universe (with time as the third dimension) by observing billions of galaxies out to 10 billion light-years, across more than a third of the sky.

ESA Euclid Telescope

GCC

Genomics Coordination Centre, UMCG/UG

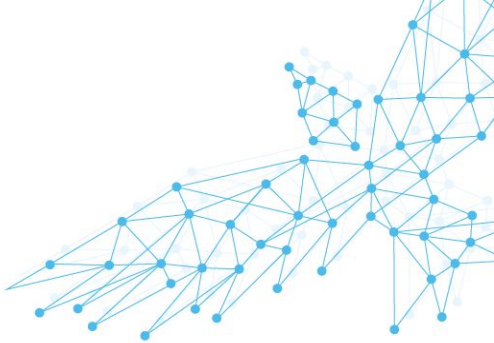
>3000 Individual Researchers
(including you!)

100+ Publications/year

Current Tally:
1306
since 2013

2025

1. Aróstica, R., Nolte, D., Brown, A., et al. (2025). A software benchmark for cardiac elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, *435*, 117485 [DOI | http]
2. Bai, S., Yang, X., Koszorús, Á., et al. (2025). Charge Radii of Neutron-Rich Scandium Isotopes and the Seniority Symmetry in the 0 f 7 / 2 Shell. *Physical Review Letters*, *134*(18), 182501 [DOI | http]
3. Brait, N., Hackl, T., & Lequime, S. (2025). detectEVE: Fast, Sensitive and Precise Detection of Endogenous Viral Elements in Genomic Data. *Molecular Ecology Resources*, *25*(4), e14083 [DOI | http]
4. Bravo, H., Ben Chehida, Y., & Van Der Meij, S. E. T. (2025). SAMPLE: An R Package to Estimate Sampling Effort for Species' Occurrence Rates. *Ecology and Evolution*, *15*(4), e70998 [DOI | http]
5. Bruegger, J., Cătană, D. I., Macovaz, V., et al. (2025). Large-Image Object Detection for Fine-Grained Recognition of Punches Patterns in Medieval Panel Painting. In P. Machado, C. Johnson, & I. Santos, editors, *Artificia* Nature Switzerland, Cham. ISBN 978-3-031-90166-9 978-3-031-90167-6. Series Title: Lecture Notes in Computer Science [DOI | http]
6. Brunner, F., Andric, P., & Maresca, F. (2025). An atomistic *K* -test framework for general grain boundaries and triclinic single crystals. *Modelling and Simulation in Materials Science and Engineering*, *33*(3), 035004 [DO
7. Chik, H. Y. J., Sibma, A., Mannarelli, M.-E., et al. (2025). Heritability and age-dependent changes in genetic variation of telomere length in a wild house sparrow population. *Evolution Letters*, *9*(2), 209–220 [DOI | h
8. Cotteret, M., Greatorex, H., Renner, A., et al. (2025). Distributed representations enable robust multi-timescale symbolic computation in neuromorphic hardware. *Neuromorphic Computing and Engineering*, *5*(1), 014008
9. Cui, J., Olthof, M., Hasselman, F., & Lichtwarck-Aschoff, A. (2025). Examining the research methods of early warning signals in clinical psychology through a theoretical lens. *BMC Psychiatry*, *25*(1), 261 [DOI | http
10. De Biase, A., Sijtsema, N. M., Van Dijk, L. V., et al. (2025). Uncertainty-aware deep learning for segmentation of primary tumor and pathologic lymph nodes in oropharyngeal cancer: Insights from a multi-center cohort. *C*
11. Dimastrogiovanni, E., Fasiello, M., Michelotti, M., & Özsoy, O. (2025). A universal constraint on axion non-Abelian dynamics during inflation. *Journal of Cosmology and Astroparticle Physics*, *2025*(03), 007 [DOI | htt
12. Driver, M. D. & Onck, P. R. (2025). Selective phase separation of transcription factors is driven by orthogonal molecular grammar. *Nature Communications*, *16*(1), 3087 [DOI | http]
13. Duijnstee, D. R., Tromp, M., Browne, W. R., & Staykov, A. (2025). Mobility of thiolates on Au(111) surfaces. *Physical Chemistry Chemical Physics*, *27*(9), 4892–4904 [DOI | http]
14. Erven, J. A. M., Mattiangeli, V., Dreshaj, M., et al. (2025). Archaeogenomic insights into commensalism and regional variation in pig management in Neolithic northwest Europe. *Proceedings of the National Academy of Sc*
15. Espinoza Cangahuala, M. K., Krishnaswamy, S. R., Kuevda, A. V., Pshenichnikov, M. S., & Jansen, T. L. C. (2025). The first step of cyanine dye self-assembly: Dimerization. *The Journal of Chemical Physics*, *162*(5), 0543
16. Fellert, M., Hein, R., Ryabchun, A., et al. (2025). A Multiresponsive Ferrocene-Based Chiral Overcrowded Alkene Twisting Liquid Crystals. *Angewandte Chemie International Edition*, *64*(1), e202413047 [DOI | http]
17. Ferrao, J. L., Mikaelson, L., Pepper, K., & Perez-Campanero, N. (2025). Self-Ablating Transformers: More Interpretability, Less Sparsity. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications* [
18. Gulyaeva, A., Liu, L., Garmaeva, S., et al. (2025). Identification and characterization of *Faecalibacterium* prophages rich in diversity-generating retroelements. *Microbiology Spectrum*, *13*(2), e01066–24 [DOI | http
19. Hein, R., Gisbert, Y., & Feringa, B. L. (2025). Multi-State Redox and Light-Driven Switching of Pseudorotaxanation and Cation Shuttling. *Journal of the American Chemical Society*, *147*(16), 13649–13657 [DOI | http
20. Huitema, I. R., Oskooei, A. R., Aktaş, M. S., & Riveni, M. (2025). Investigating Echo-Chambers in Decentralized Social Networks: A Mastodon Case Study. In H. Cherifi, M. Donduran, L. M. Rocha, C. Cherifi, & O. Varol, e Nature Switzerland, Cham. ISBN 978-3-031-82430-2 978-3-031-82431-9. Series Title: Studies in Computational Intelligence [DOI | http]
21. Jansen, T. L. C. (2025). The Manhattan exciton size: A physically tractable delocalization measure. *The Journal of Chemical Physics*, *162*(7), 074113 [DOI | http]
22. Kolbe, M., Bintanja, R., Van Der Linden, E. C., & Cordero, R. R. (2025). Vertical structure and surface impact of atmospheric rivers reaching antarctic sea ice and land. *Atmospheric Research*, *315*, 107841 [DOI | ht
23. Kolbe, M., Torres Alavez, J. A., Mottram, R., et al. (2025). Model performance and surface impacts of atmospheric river events in Antarctica. *Discover Atmosphere*, *3*(1), 4 [DOI | http]
24. Kuevda, A. V., Espinoza Cangahuala, M. K., Hildner, R., Jansen, T. L. C., & Pshenichnikov, M. S. (2025). Linear Dichroism Microscopy Resolves Competing Structural Models of a Synthetic Light-Harvesting Complex. *Jou*
25. Kyuberis, A. A., Pašteka, L. F., Eliav, E., et al. (2025). Theoretical determination of the ionization potentials of ScF, YF, LaF, and AcF. *Physical Review A*, *111*(4), 042823 [DOI | http]
26. Lambert, J. W., Roeble, L., Pannetier, T., Etienne, R. S., & Valente, L. (2025). Using phylogenetic data for island biogeography analyses: The DAISIEprep package. *Molecular Phylogenetics and Evolution*, *206*, 108324 [
27. Ma, B., De Biase, A., Guo, J., et al. (2025). The prognostic value of pathologic lymph node imaging using deep learning-based outcome prediction in oropharyngeal cancer patients. *Physics and Imaging in Radiation Onc*
28. Mahajna, A., Geurkink, B., Gacesa, R., et al. (2025). Metatranscriptomes of activated sludge microbiomes from saline wastewater treatment plant. *Scientific Data*, *12*(1), 348 [DOI | http]
29. Mohamed Yunus, R. A., Gürel, U., Guzik, A., et al. (2025). Glass Transition and Yielding of Ultrasoft Charged Spherical Micelles. *Macromolecules*, page acs.macromol.4c03215 [DOI | http]
30. Mol, F. N., Van Der Hoek, L., Ma, B., et al. (2025). MRI-Based Head and Neck Tumor Segmentation Using nnU-Net with 15-Fold Cross-Validation Ensemble. In K. A. Wahid, C. Dede, M. A. Naser, & C. D. Fuller, editors, *He* 190. Springer Nature Switzerland, Cham. ISBN 978-3-031-83273-4 978-3-031-83274-1. Series Title: Lecture Notes in Computer Science [DOI | http]
31. Ndung'u, S., Grobler, T., Wijnholds, S. J., & Azzopardi, G. (2025). Content-based image retrieval using COSFIRE descriptors with application to radio astronomy. *Monthly Notices of the Royal Astronomical Society*, *537*(4)
32. Neh, H., De Vette, S., Stoffers, R., et al. (2025). Normal tissue complication probability model predicting taste impairment in head and neck cancer patients: Evaluating the taste bud bearing tongue mucosa as a predictor
33. van Noord, R., Esplà-Gomis, M., Chichirau, M., Ramírez-Sánchez, G., & Toral, A. (2025). Quality Beyond A Glance: Revealing Large Quality Differences Between Web-Crawled Parallel Corpora. In O. Rambow, L. Wanne *31st International Conference on Computational Linguistics*, pages 1824–1838. Association for Computational Linguistics, Abu Dhabi, UAE [http]
34. Ovalle, M., Doellerer, D., & Feringa, B. L. (2025). Azobenzene-Oxindole Photochromic Dyads. *Angewandte Chemie*, page e202501872 [DOI | http]
35. Qin, T., Valente, L., & Etienne, R. S. (2025). Impact of evolutionary relatedness on species diversification and tree shape. *Journal of Theoretical Biology*, *598*, 111992 [DOI | http]
36. Shaheen, I., Brakman, S., Canello, J., & Garretsen, H. (2025). Firm interconnectedness and resilience: evidence from the Italian manufacturing. *Cambridge Journal of Regions, Economy and Society*, *18*(1), 211–226 [
37. Stindt, C. N., Jo, T., Steen, J. D., Feringa, B. L., & Crespi, S. (2025). Computational Study on the Dynamics of a Bis(benzoxazole)-Based Overcrowded Alkene. *The Journal of Physical Chemistry A*, *129*(5), 1301–1309 [
38. Stoica, A. & Riveni, M. (2025). Provenance for Longitudinal Analysis in Large Scale Networks. In L. M. Aiello, T. Chakraborty, & S. Gaito, editors, *Social Networks Analysis and Mining*, volume 15212, pages 274–285. Sprir Lecture Notes in Computer Science [DOI | http]
39. Tang, T., Snijders, T., & Flache, A. (2025). An Empirical and Simulation Investigation of Bounded Confidence and Negative Influence in Opinion Dynamics Using Stochastic Actor-Oriented Model. *Journal of Artificial Societ*
40. Tsakali, N., Kolbe, M., Bintanja, R., & Skyllas, N. (2025). The time of emergence of Arctic warming, wetting and sea ice melting. *Scientific Reports*, *15*(1), 12626 [DOI | http]
41. Wang, J., Cai, J., Tang, W., et al. (2025). A comparison of an integrated and image-only deep learning model for predicting the disappearance of indeterminate pulmonary nodules. *Computerized Medical Imaging and Gr*
42. Wang, X. (2025). EGOFALLS: A Visual-Audio Dataset and Benchmark for Fall Detection Using Egocentric Cameras. In A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, & U. Pal, editors, *Patter*



What do we have today?

university of groningen / center for information technology

Hábrók

20256 CPU Cores
204 Machines
60 GPUs
13PB Storage

1250+
Software
Packages

Clouds

Bateleur:
25144 CPU cores
137 TB RAM
206 TiB VM Storage
4 PB Filesystem
Hosts Hábrók
Bare metal focused

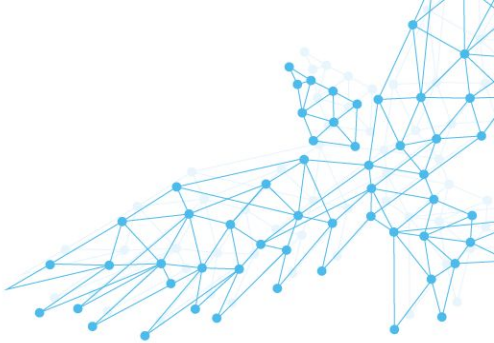
Merlin:
1959 CPU cores
9.6 TB RAM
439 TiB VM Storage
3.2 PB Filesystem
Virtualisation focused

Hábrók (PaaS)

Cloud Platform (IaaS)

university of groningen / center for information technology

Technologies we use for IaaS today



- OpenStack
  - Virtual Machines (QEMU/kvm)
  - Bare Metal
  - Virtual Networking (OpenVSwitch)

- Centralised Storage:
  - Ceph
    - Block Storage
  - Lustre
    - Filesystems

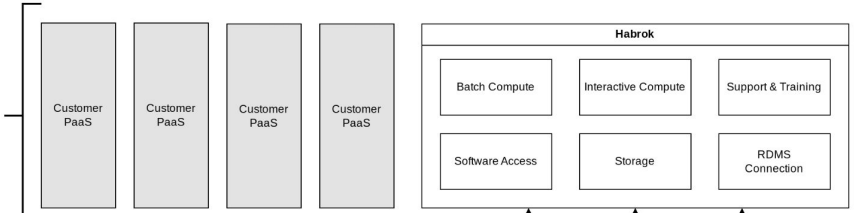
university of groningen / center for information technology

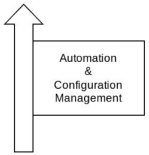
Our Design

Structured around NIST 800-145 Definition of Cloud Computing

Platform as a Service

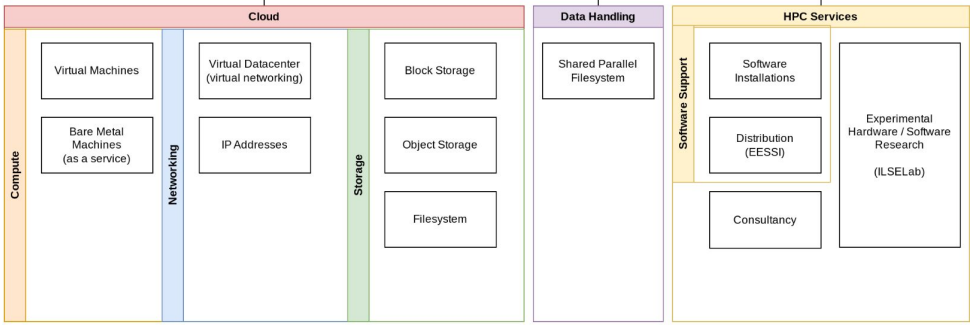
Our building blocks come together to form platforms that our users can do research on.

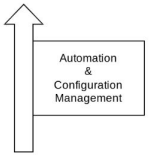




Infrastructure as a Service

We provide a toolkit for those that want to build their own platforms, made of generic building blocks.





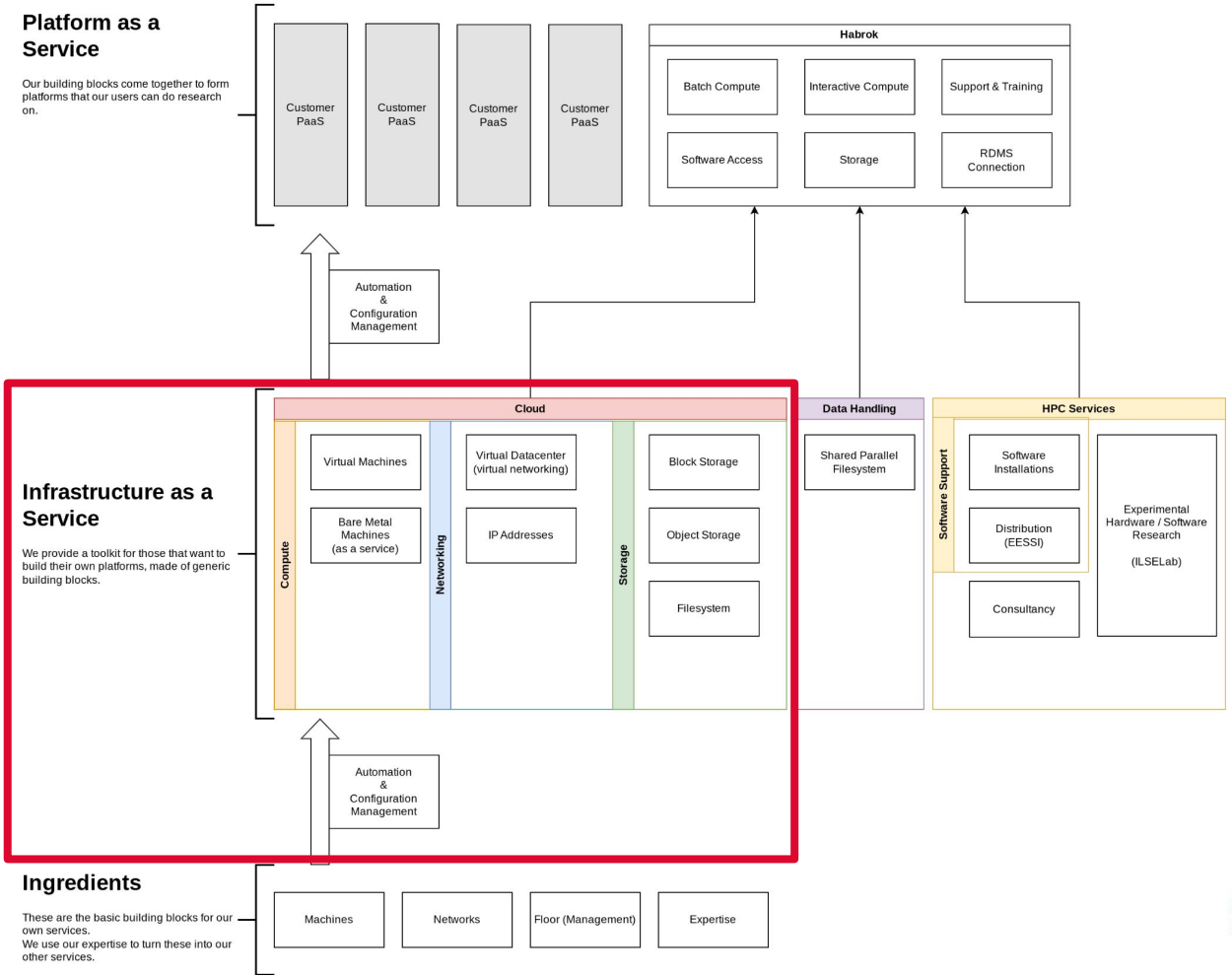
Ingredients

These are the basic building blocks for our own services.
We use our expertise to turn these into our other services.

center for information technology

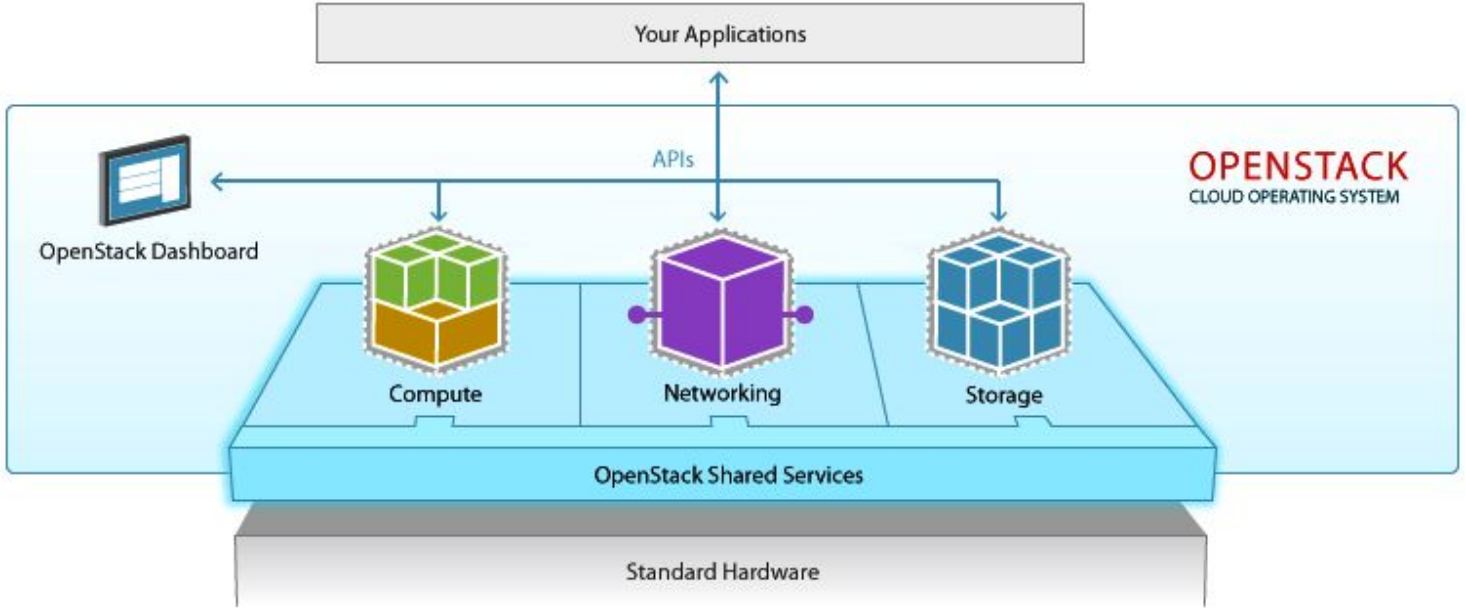
Our Design

Structured around NIST 800-145 Definition of Cloud Computing



center for information technology

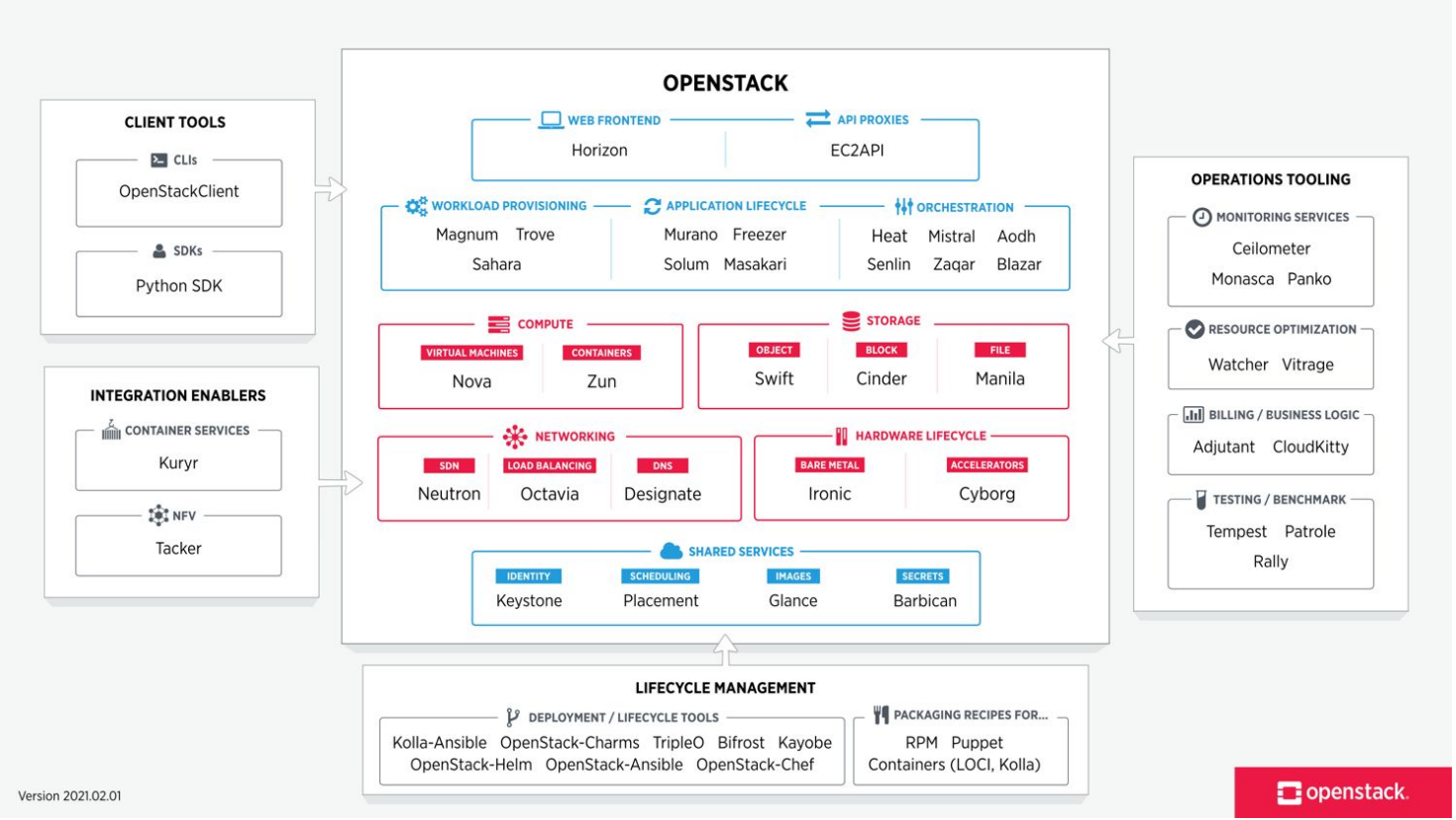
OpenStack



university of groningen / center for information technology

Also OpenStack

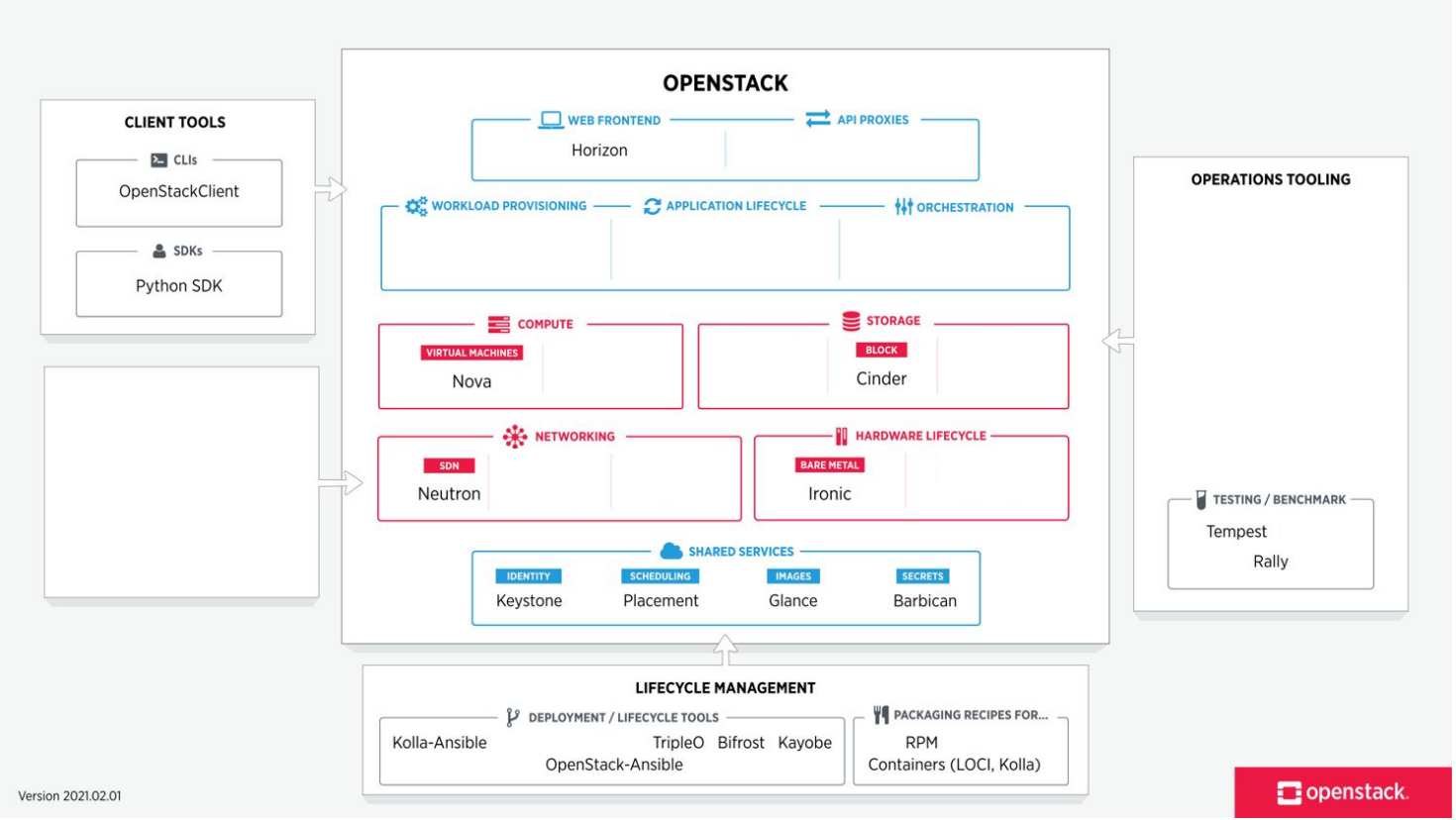
(we don't need **all** of this)



university of groningen / center for information technology

Also OpenStack

(what we **actually** use)

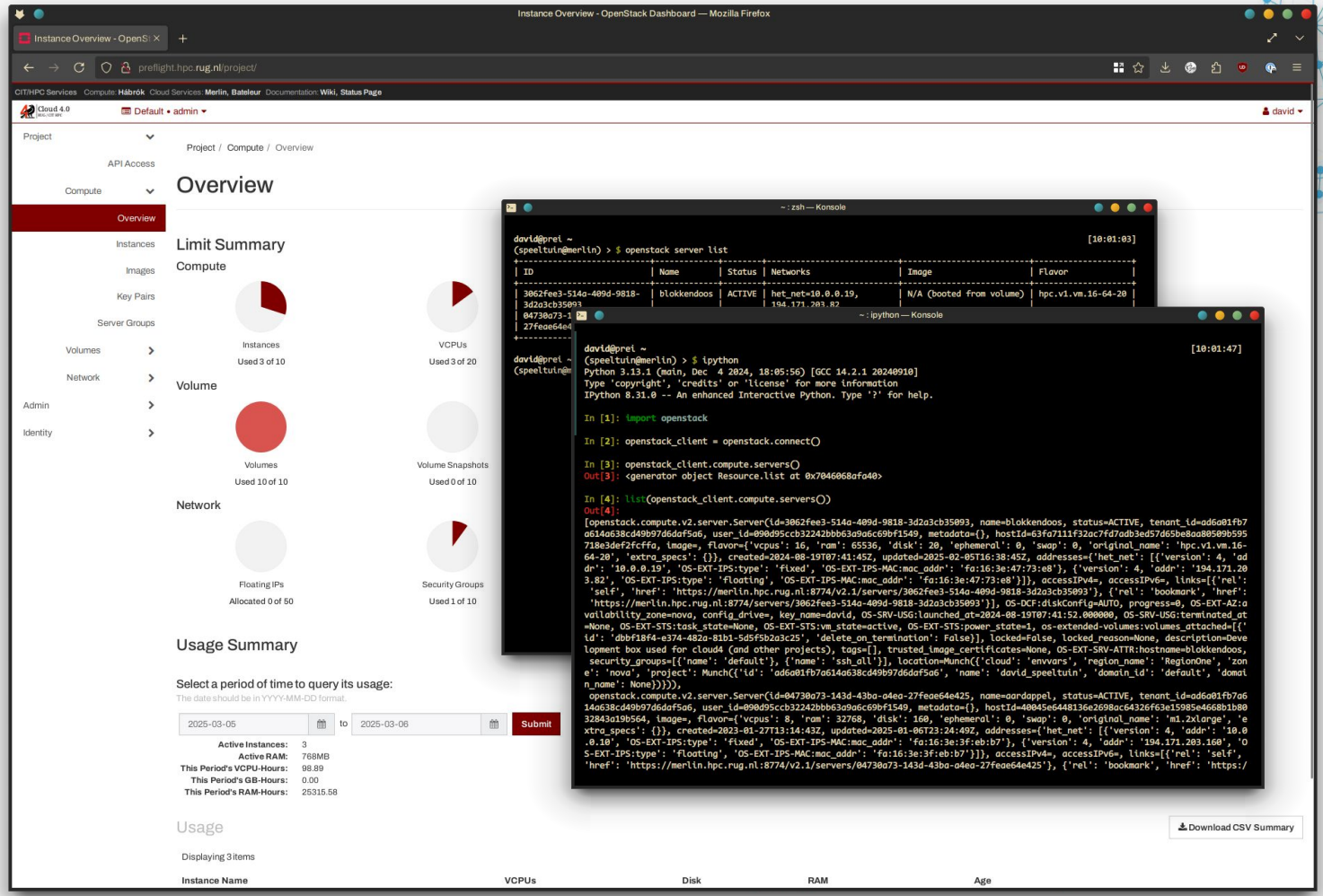


university of groningen / center for information technology

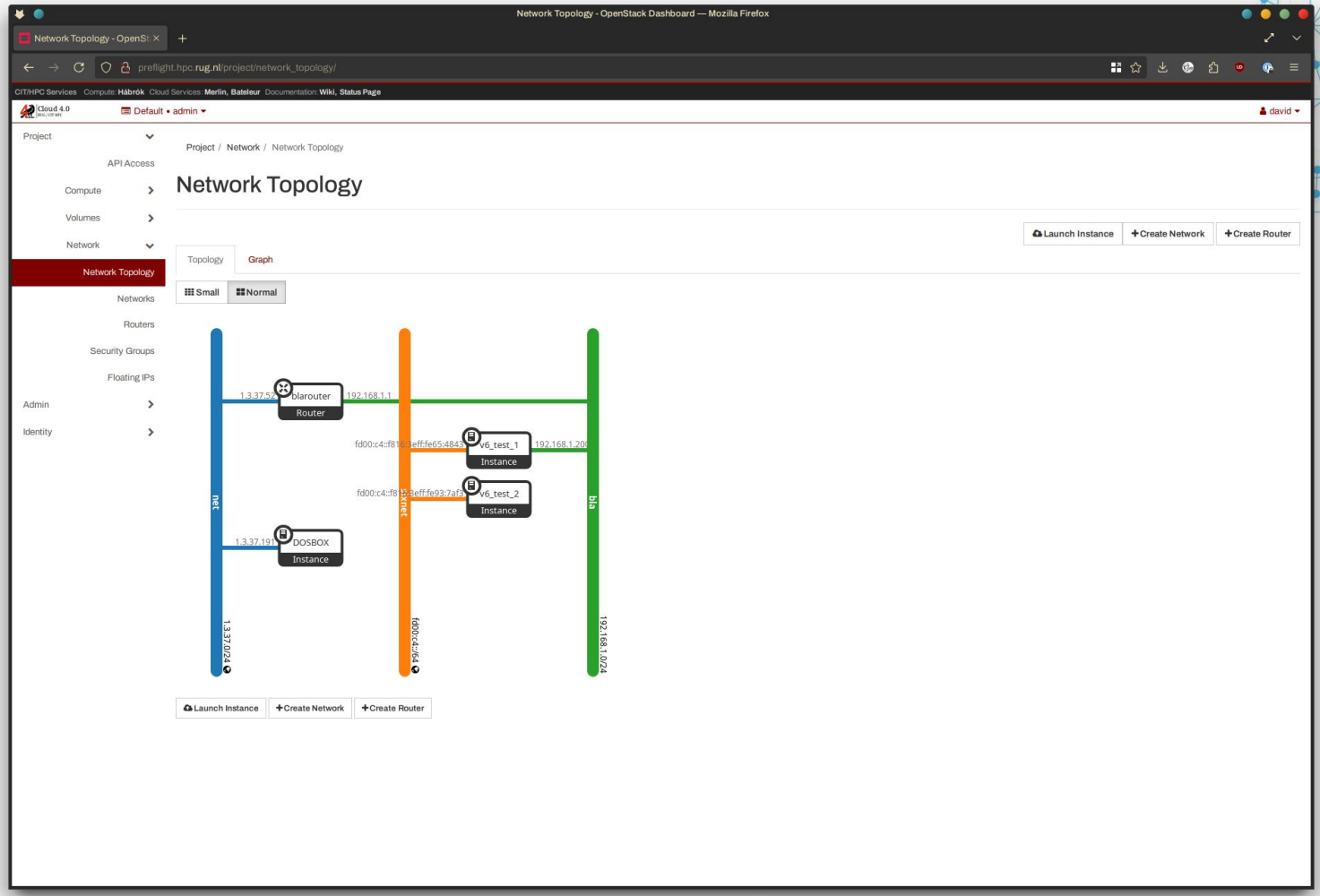
Openstack

One API

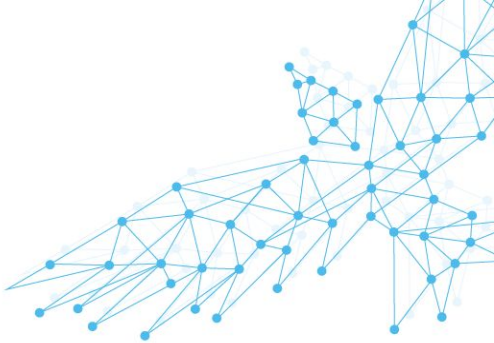
Many Clients



Virtual Networks

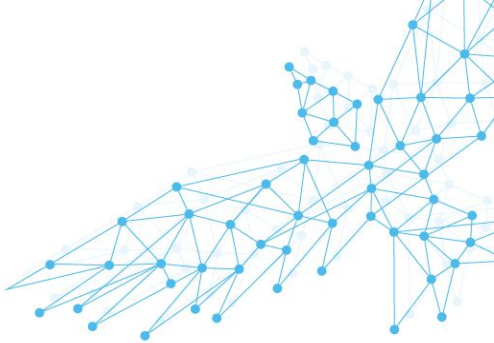


How do we deploy it today?



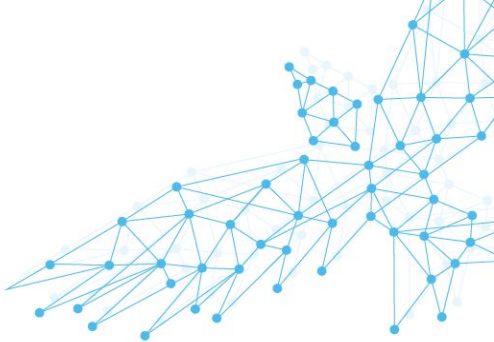
- Use the OpenStack official deployment tooling
  - Kolla
  - Kolla-Ansible
  - Kayobe
- Undercloud/Overcloud Architecture
  - Build a big cloud with a small cloud
- For Ceph
  - cephadm
  - Kayobe

Challenges



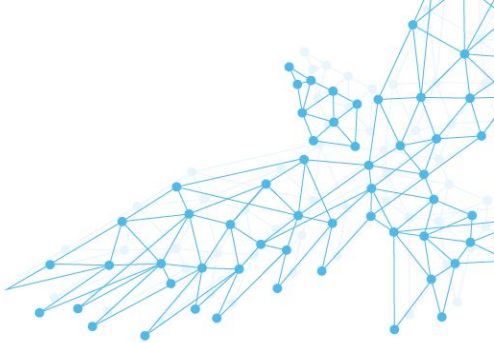
- Existing stack is incredibly complex
  - many interconnected moving parts
  - makes maintenance/evolution very labor intensive

- Dependent on external deployment system
  - different priorities
  - contains lots of things we don't need, but that can break

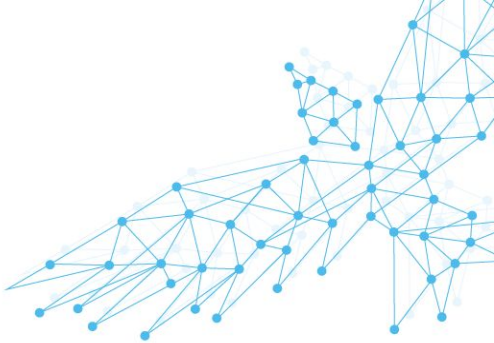


The way systems grow organically does not lead to the most optimal setup

Something better?



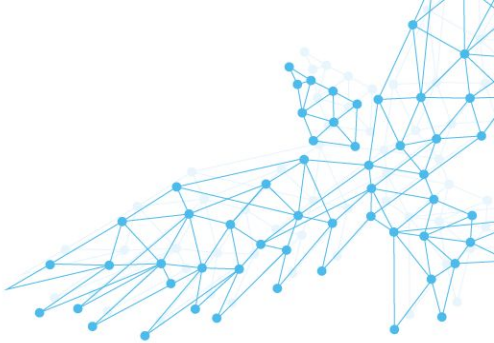
- Make maintenance easier
  - Improve stability, reliability, and the lives of our engineers
- Determine for ourselves what we do and don’t want in our infrastructure
  - Less is more
- Independence
  - We determine what is important for our stack



Where do we go from here?

university of groningen / center for information technology

So why do we do this ourselves?



No other options:

- Off the shelf appliances
- External hosting
- Public Cloud

Common pattern:

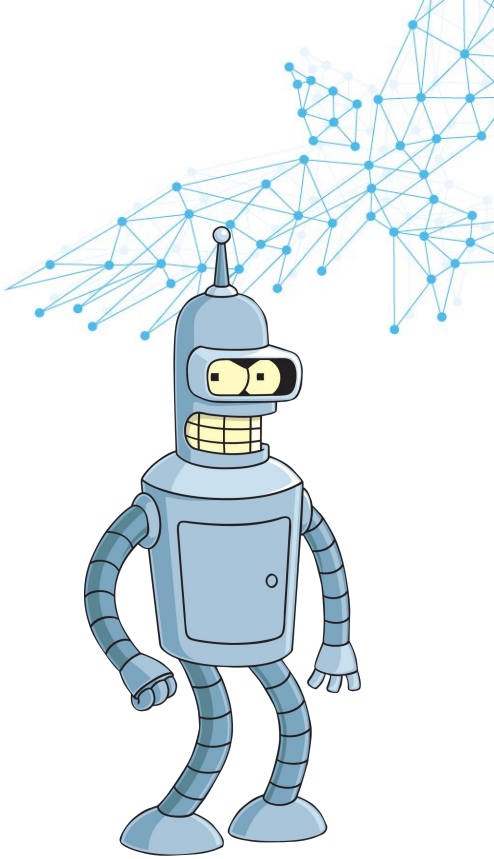
- Stuff you don’t need
- for too much money,
- with vendor lock-in,
- and with little autonomy.

- Also, tendering is not fun

university of groningen / center for information technology

New Approach

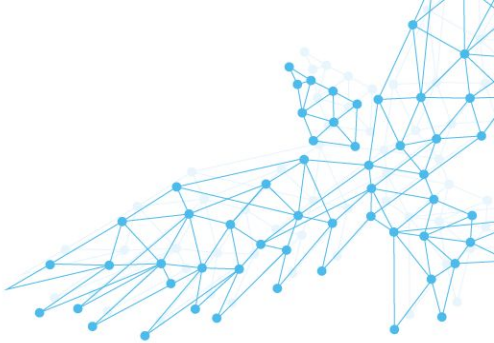
- Our own deployment (distro) of OpenStack
- Built with Salt / Python
- Apply YAGNI
- Core principle:
  Everything Should Be Made as Simple as Possible, But Not Simpler



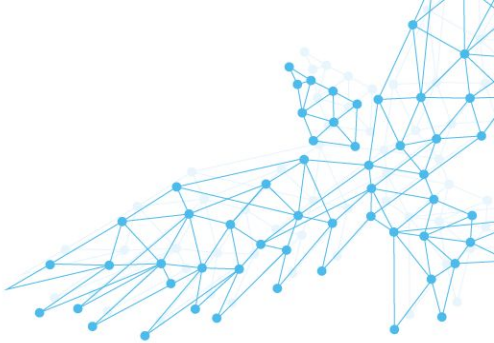
university of groningen / center for information technology

Goals

- Maintain / evolve institutional knowledge
- Stable / predictable environments
- Optimised for lifecycle work, not deployment work



How will we manage all this?

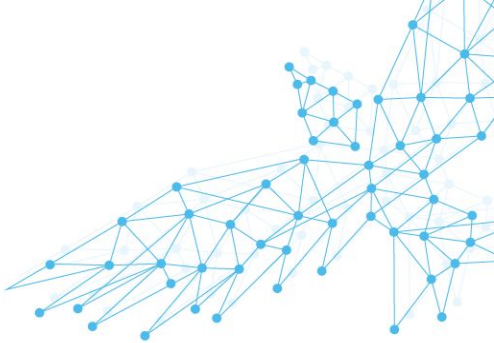


- Infrastructure as Code is a **must**
- Heavily prefer open source technologies
  - And contribute when we can
- Work with the garage door open, using a public git repo

What's next?

- Building a new platform to replace Bateleur / Merlin: **Cloud4**
- How to start?
  - **Strong design principles**

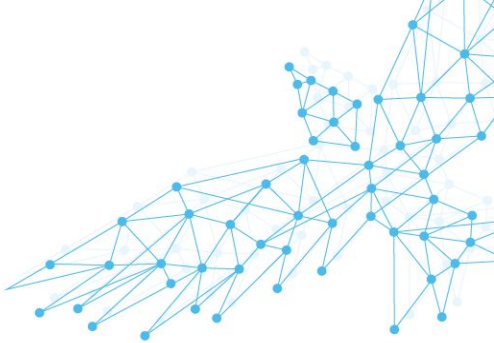
university of groningen / center for information technology



Our 4 design principles

university of groningen / center for information technology

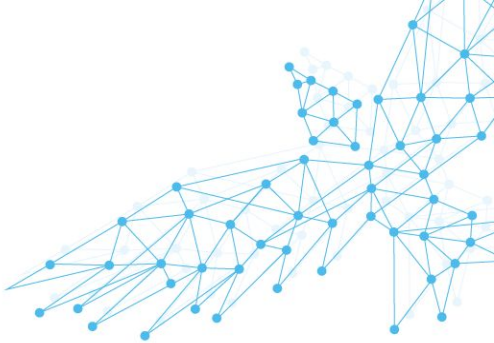
1. Opinionated framework



- **Cloud4** is our new framework,
  aimed at our own case/priorities.

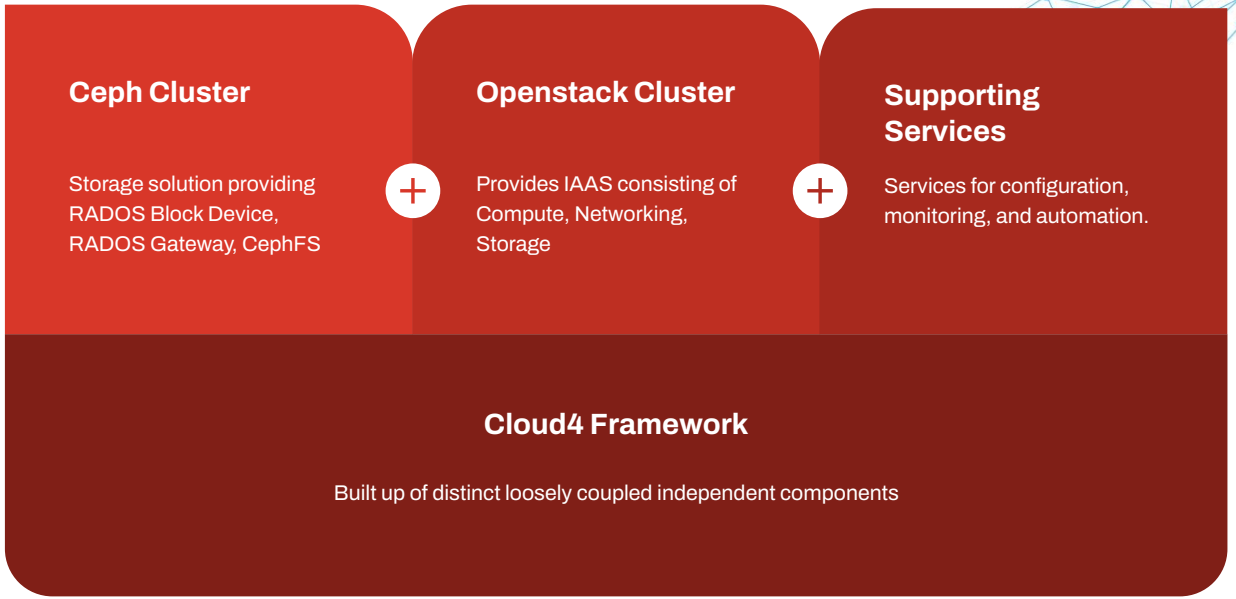
- We use that framework to build cloud(s)

1. Opinionated framework



- One size fits one
- Problem-First, not Cool-Tech-First (as boring as possible)
- Minimize complexity

The Cloud Design



university of groningen / center for information technology

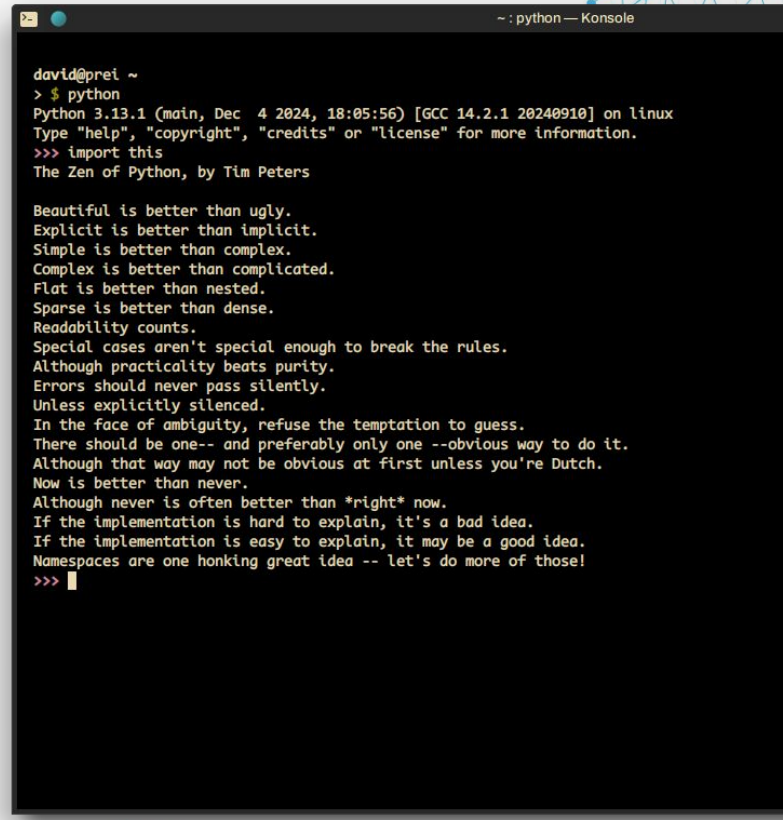
2. Focus on the lifecycle

- 79% of costs over the lifetime of a system is “lifecycle costs”[1]
- Maintenance is more important than deployment
- All normal work should be automated or automatable

[1] “Zarnekow, Ruediger, and Walter Brenner. ‘Distribution of Cost over the Application Lifecycle - a Multi-Case Study.’, 1 January 2005, 79

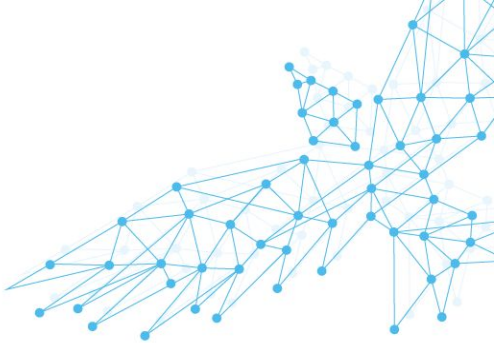
3. Have ONE way of doing things

- “There should be one – and preferably only one – obvious way to do it.”
  - Tim Peters; The Zen of Python



3. Have ONE way of doing things

- For us this means:
  - The code's structure should encourage the correct behaviour
  - Declarative Config Management
  - The Exception is the Rule

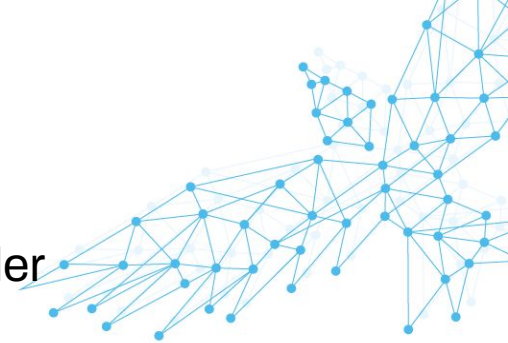


Exception is the rule: an example

pillar : nvim — Konsole

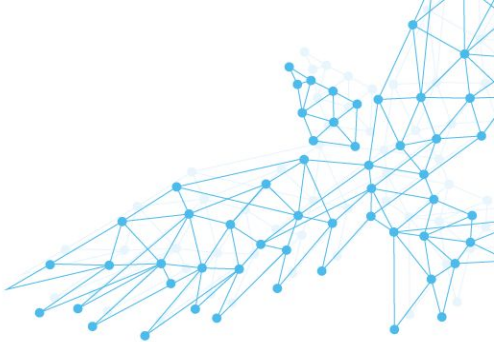
```
22
23 def run():
24     """
25     Renders the base environment pillar assignment.
26     Assigns the minions their pillar file based on the name they have.
27     """
28     minion_id = __opts__["id"]
29
30     # We use the fqdn as the default minion_id, so splitting on
31     # the first . gets us the plain hostname
32     hostname, domainname = minion_id.split(".", 1)
33     if "-" in hostname:
34         # We have multiple of the same type
35         hostname, count = hostname.rsplit("-", 1)
36     else:
37         count = 0
38     # Replace the other . in domainnames with _
39     # That makes filenames a lot easier
40     domainname = domainname.replace(".", "_")
41
42     relevant_pillar_files = []
43
44     add_if_pillar_file_exists(relevant_pillar_files,  "defaults.sls")
45     add_if_pillar_file_exists(relevant_pillar_files, f"defaults/{hostname}.sls")
46     add_if_pillar_file_exists(relevant_pillar_files, f"defaults/{hostname}/{count}.sls")
47     add_if_pillar_file_exists(relevant_pillar_files, f"environments/{domainname}/defaults.sls")
48     add_if_pillar_file_exists(relevant_pillar_files, f"environments/{domainname}/{hostname}.sls")
49     add_if_pillar_file_exists(relevant_pillar_files, f"environments/{domainname}/{hostname}/{count}.sls")
50
51
52     pillar_top = {"base": {"*": relevant_pillar_files}}  # noqa: 206
53
54     return pillar_top
```

4. Complexity

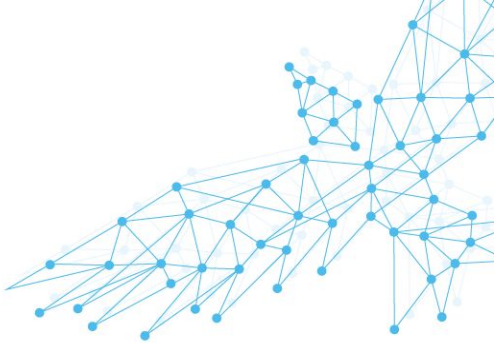


- Everything should be as simple as possible, but no simpler
- Complexity does not disappear when hidden (in other systems/libraries)
- Apply YAGNI to keep complexity in check
- Delete, Delete, Delete

Principles



1. Build an opinionated framework, then build your platform
2. Focus on the lifecycle, not the deployment
3. Have ONE way of doing things, and have that be the most intuitive option
4. Prefer simple over complex, complex over complicated



What does this all get us?

university of groningen / center for information technology

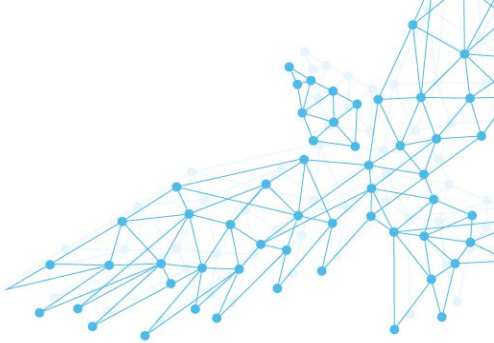
Main Differences

Old	Cloud4
Has to work for everyone	Not generic
Static Configuration	Dynamic Configuration
299490 LOC	5519 LOC (<2%)
Lots of tech debt	Easier to add functionality and grow
Big-Bang updates required	Can be upgraded gradually

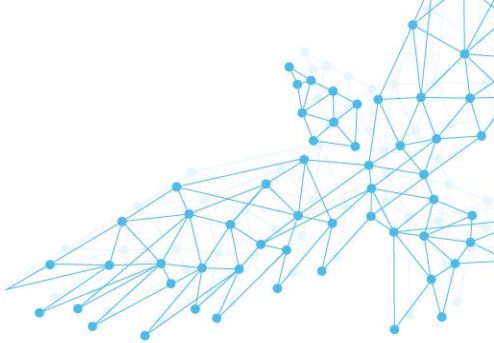
Main Differences (technical)

Old	Cloud4
Many Many Many Containers	No No No Containers
Templates in templates in ... all the way down	Just... no Clean python code or *simple* Jinja2
Control plane runs networking	Control plane separate from network traffic
VLAN-Land	Edge Routing
IPv4-only	IPv6-only internally

Cloud migration

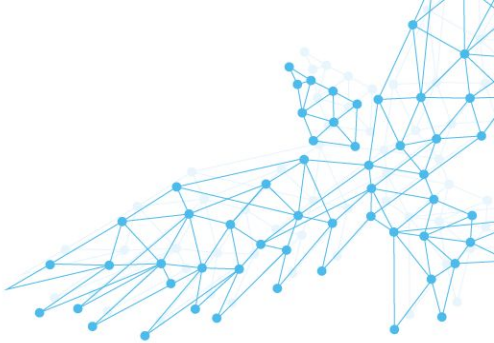


1. Deploy a new Cloud4 control plane / setup in a rack next to the old cloud
2. Clear out machines from the old cloud one-by-one
3. Introduce those machines to the new cloud
4. Migrate existing customers over to the new cloud
    a. Early adopters will want to
    b. The bulk will pragmatically follow features / qualities
    c. Laggards will feel threatened by EOL date
5. ???
6. Profit



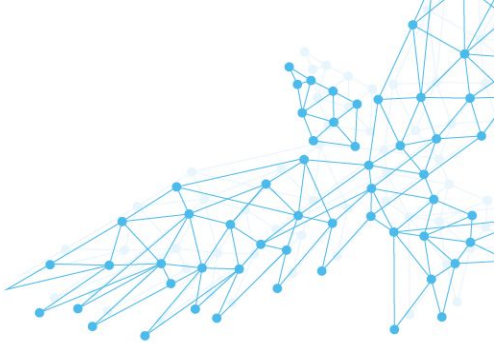
Cool, but what's the catch?

It’s a whole load of work



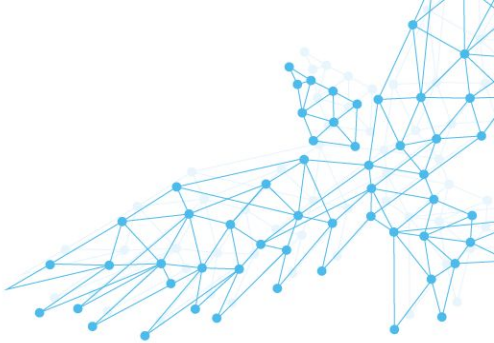
- Build from scratch
  - High amount of effort
  - Have to be in it for the long haul (1.5 years and counting)
- Discovering our requirements and constraints
- Understanding Legacy
  - Lots and lots of trial and error
  - … and software archeology

Inheriting Living Legacy



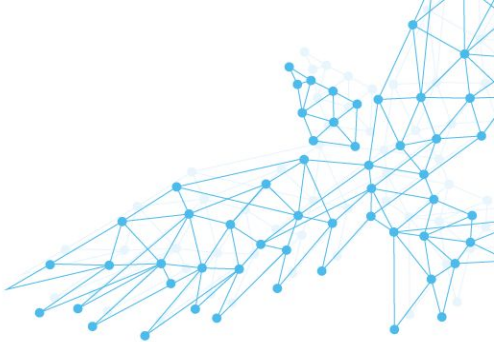
- We have to keep the old environments up and running
  - Inherited from the before times

- We have to keep the stakeholders in the loop
  - Migration path should be clear



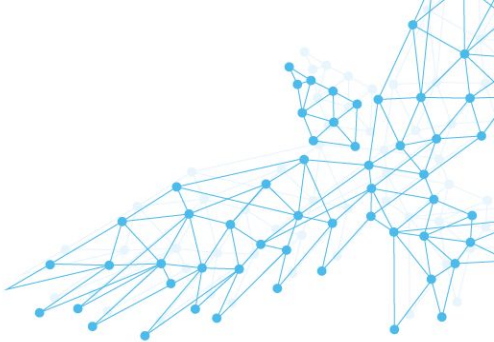
Recap

university of groningen / center for information technology



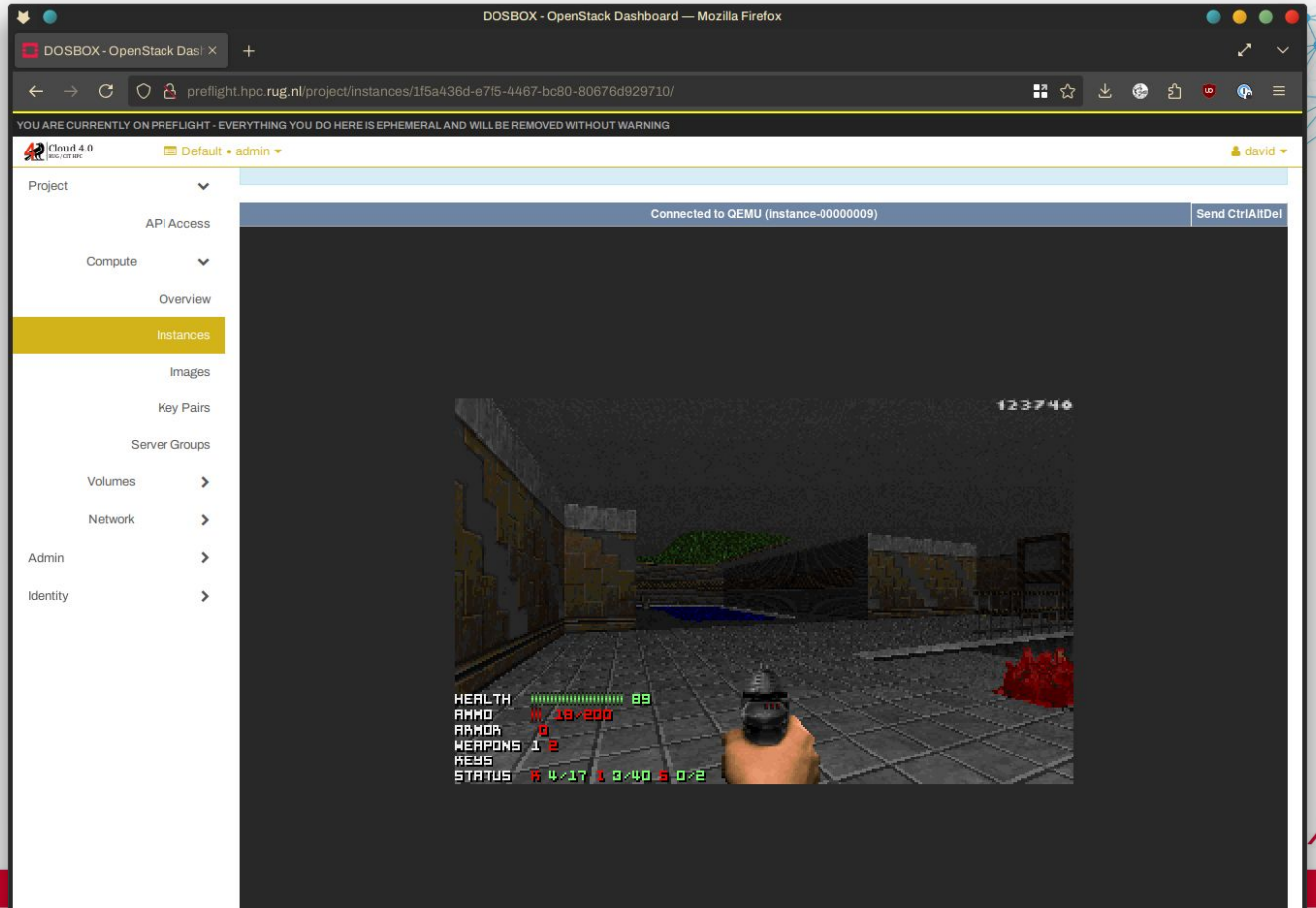
The way systems grow organically does not lead to the most optimal setup

Principles



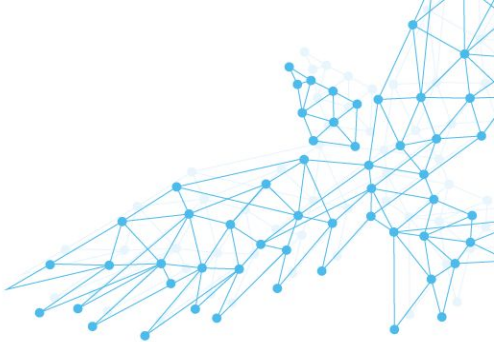
1. Build an opinionated framework, then build your platform
2. Focus on the lifecycle, not the deployment
3. Have ONE way of doing things, and have that be the most intuitive option
4. Prefer simple over complex, complex over complicated

Can it run DOOM?



center for information technology

Wanna see the code?



You can!

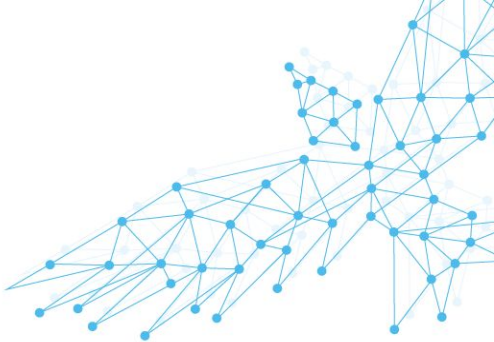
https://gitrepo.service.rug.nl/cit-hpc/cloud-4/cloud4

It works for us, and just for us.
(it's really boring)

university of groningen / center for information technology

Thanks

Contact us at hpc@rug.nl for more information/questions.



university of groningen / center for information technology