

Shifting from performance measurements to resilience measurements

Hélène Coullon¹

¹IMT Atlantique, Inria, LS2N

July 10, 2026



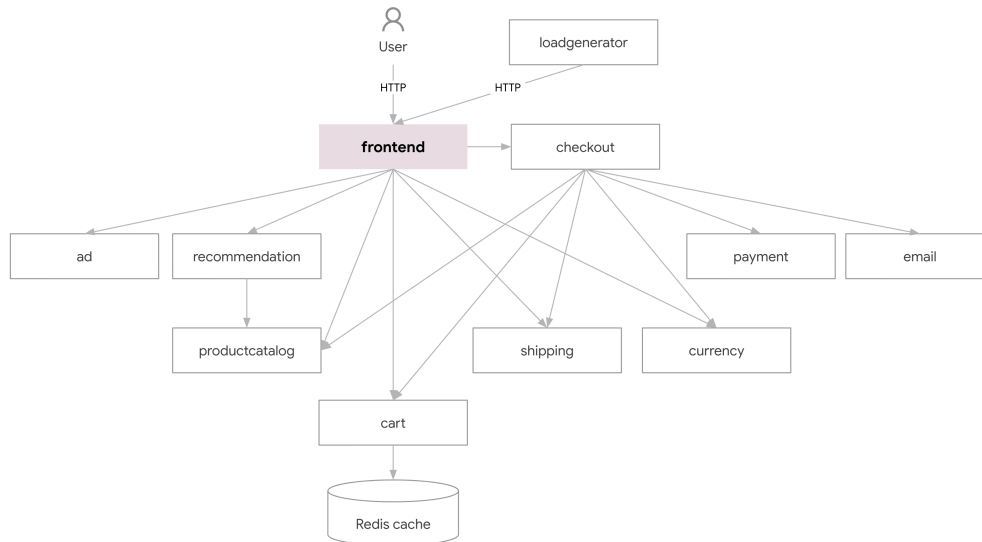
Outline

Dynamic reconfigurations of digital infrastructures	2
Evaluating dynamic reconfigurations	9
Resilience of digital infrastructures	12
Evaluating resilience?	26
Bibliography	35

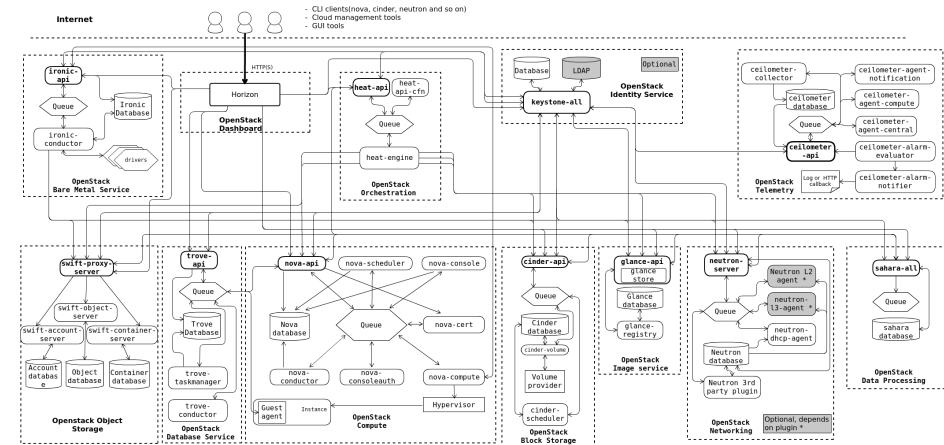
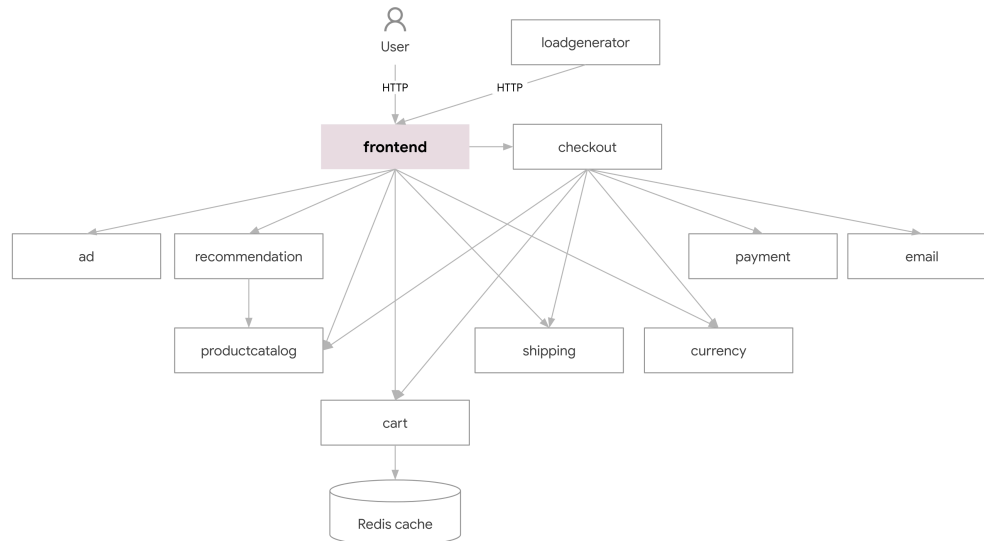
Dynamic reconfigurations of digital infrastructures

Dynamic reconfigurations

- ▶ Dynamic reconfiguration originates from *CBSE* [1]
 - models able to dynamically change the assembly of components

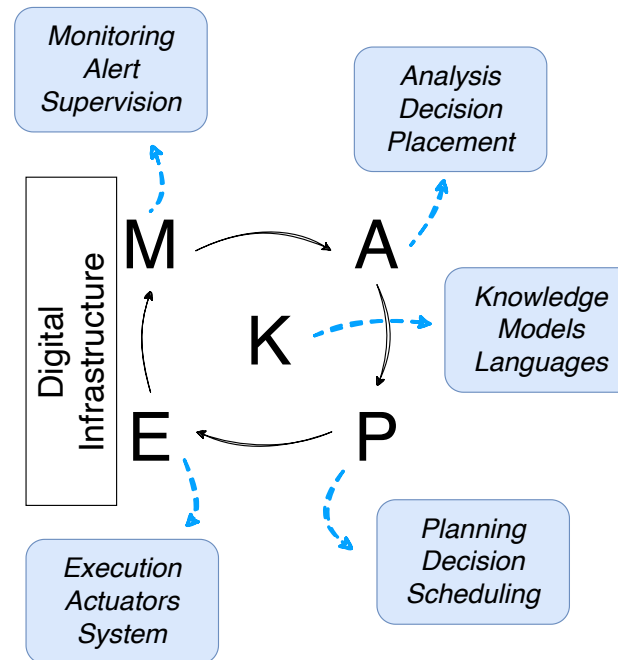


- ▶ Dynamic reconfiguration originates from *CBSE* [1]
 - models able to dynamically change the assembly of components



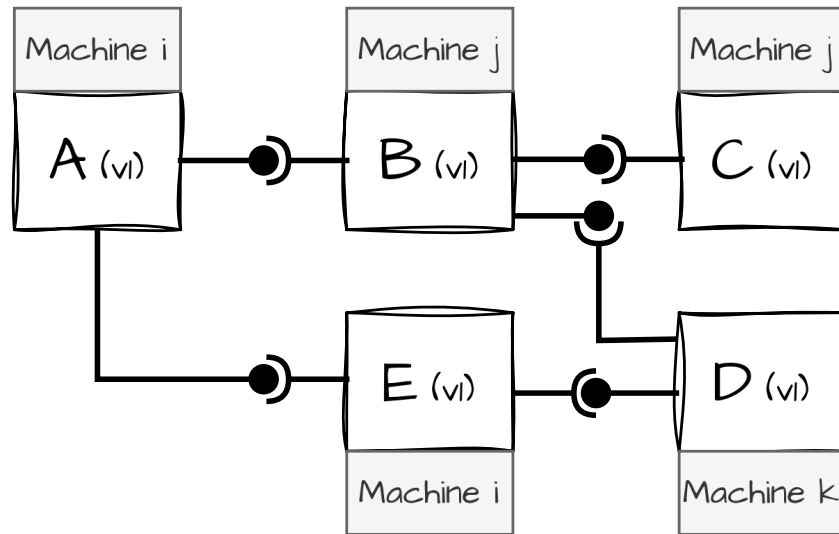
Autonomic computing

- ▶ MAPE-K adaptation model:
 - *Monitor, Analyze, Plan, Execute, Knowledge* [2]
- ▶ CBSE reconfiguration focuses on *PEK* of the adaptation loop MAPE-K [2]



Example

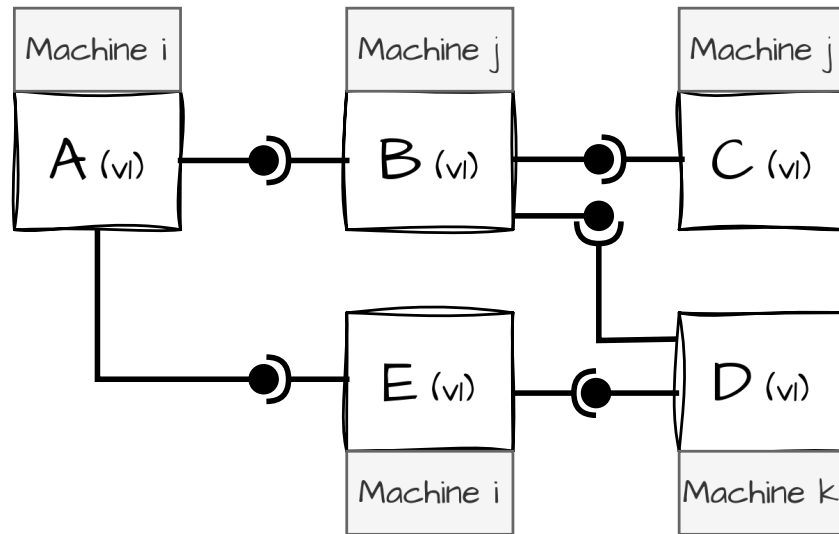
⇒ *Concrete example: update component A in the assembly*



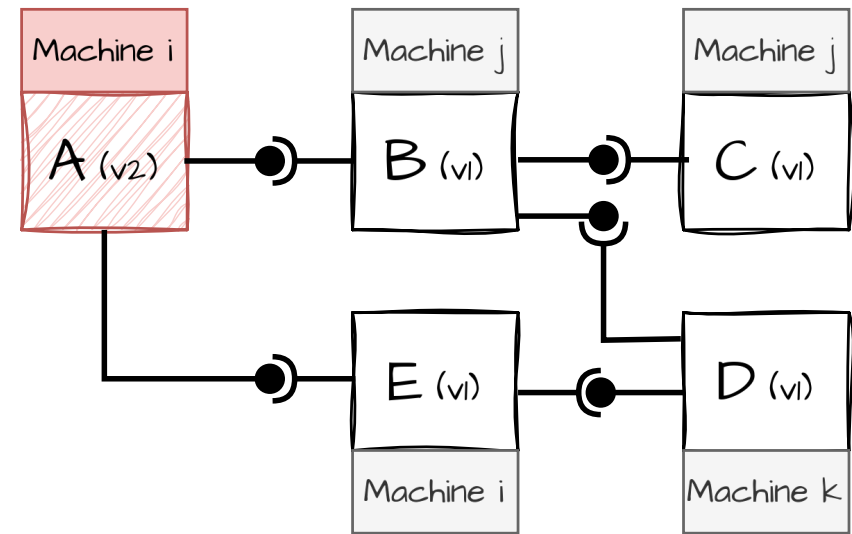
Initial state

Example

⇒ *Concrete example: update component A in the assembly*

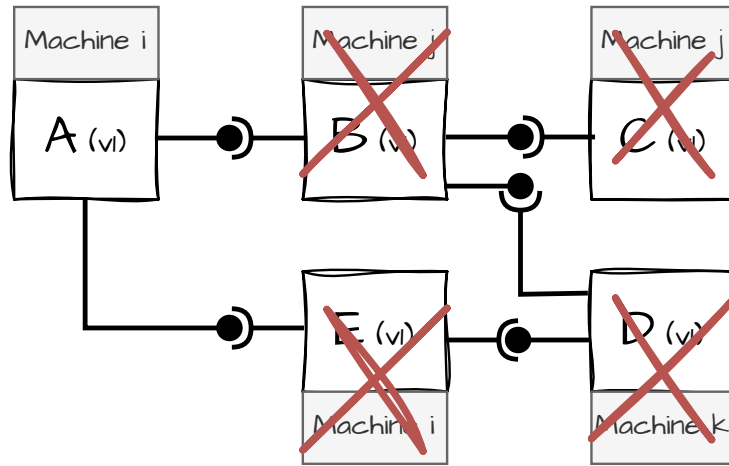


Initial state



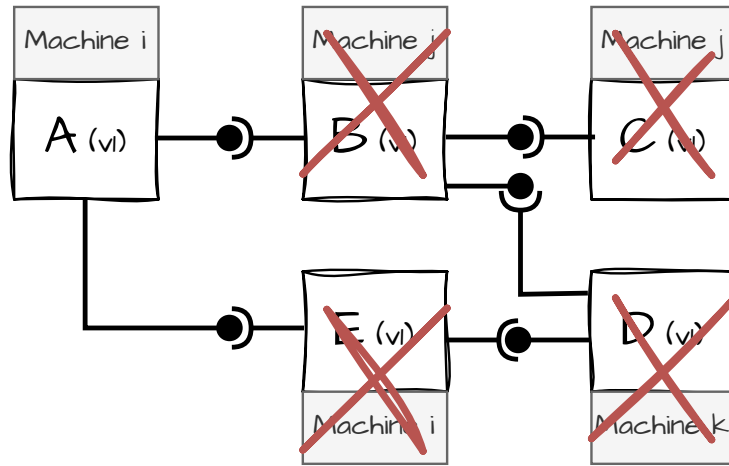
Target state

Concrete examples (downtime)

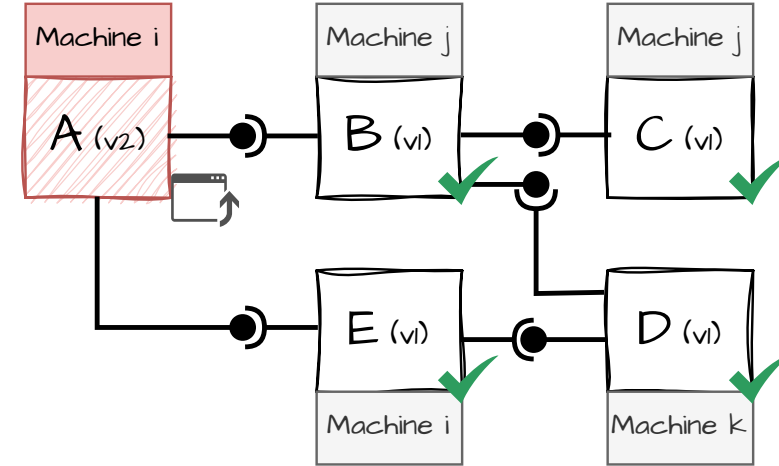


delete(D); delete(C); delete(E); delete(B);

Concrete examples (downtime)

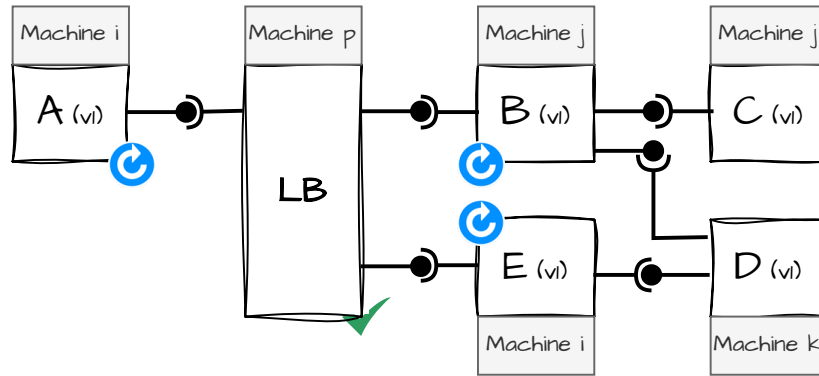


delete(D); delete(C); delete(E); delete(B);



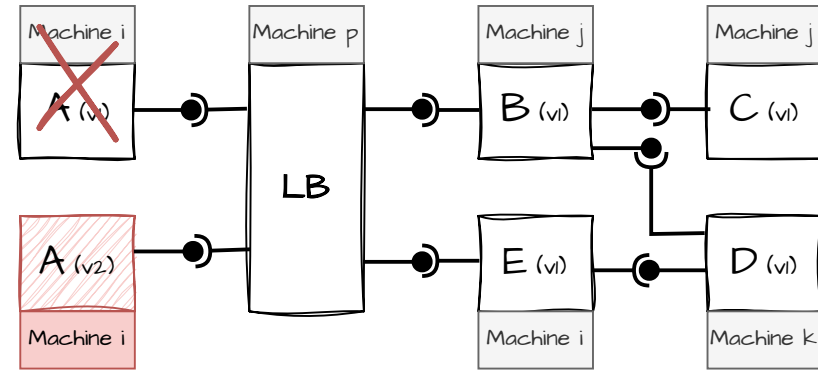
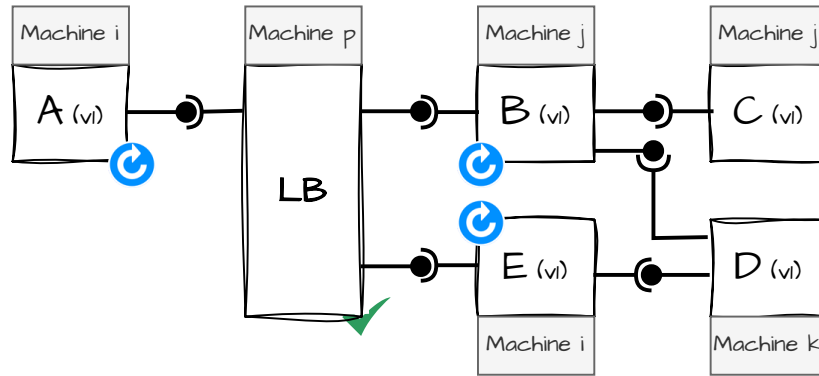
*update(A); start(B); start(E), start(C);
start(D)*

Concrete examples (A/B deployment)



start(LB); restart(B); restart(A(v1));
restart(E);

Concrete examples (A/B deployment)



*start(LB); restart(B); restart(A(v1));
restart(E);*

start(A(v2)); wait...; delete(A(v1))

⇒ **More complex lifecycle actions than version 1**

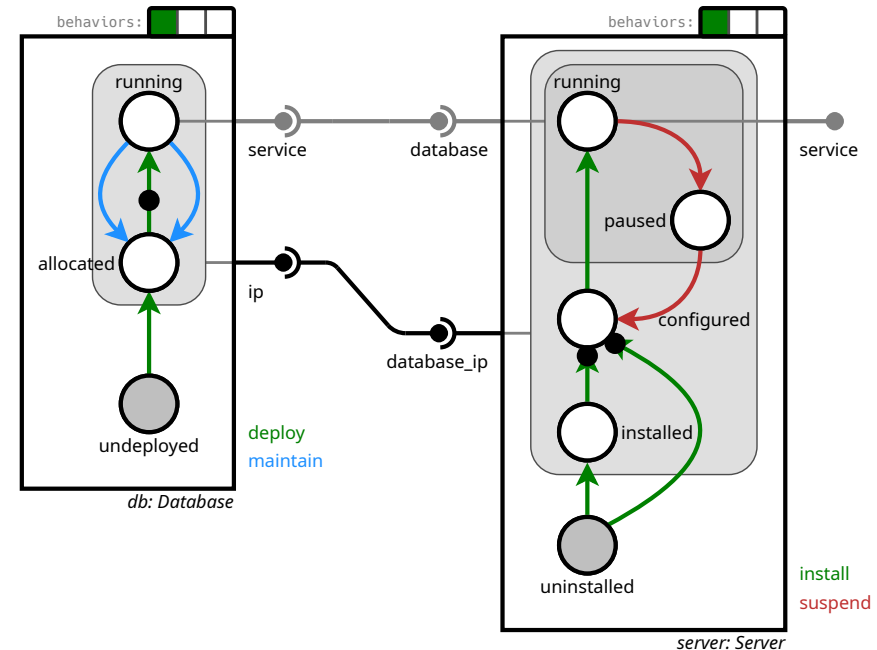
Programmable lifecycle

Safe programmable lifecycle management with Concerto 🎵 [3]

```

1 add(server: Server)
2 add(db: Database)
3 con(server.database_ip, db.ip)
4 con(server.database, db.service)
5 pushB(server, install)
6 pushB(db, deploy)
7 wait(server)
  
```

Python



- ▶ Automatic and safe coordination of programmable lifecycles
- ▶ Reconfiguration language with 6 generic instructions

Infrastructure-as-Code

Infrastructure-as-Code (IaC) meets dynamic reconfiguration

- ▶ Tools to automate and manage infrastructures through codes
- ▶ IaC embeds reconfiguration, adaptation, and lifecycle management
 - Declarative approaches

Infrastructure-as-Code

Infrastructure-as-Code (IaC) meets dynamic reconfiguration

- ▶ Tools to automate and manage infrastructures through codes
- ▶ IaC embeds reconfiguration, adaptation, and lifecycle management
 - Declarative approaches

Terraform

- ▶ PE-K
- ▶ CRUD lifecycles
- ▶ Cloud resources / APIs

Infrastructure-as-Code

Infrastructure-as-Code (IaC) meets dynamic reconfiguration

- ▶ Tools to automate and manage infrastructures through codes
- ▶ IaC embeds reconfiguration, adaptation, and lifecycle management
 - Declarative approaches

Terraform

- ▶ PE-K
- ▶ CRUD lifecycles
- ▶ Cloud resources / APIs

Kubernetes

- ▶ full MAPE-K
- ▶ CRUD lifecycles
- ▶ Containers

Infrastructure-as-Code

Infrastructure-as-Code (IaC) meets dynamic reconfiguration

- ▶ Tools to automate and manage infrastructures through codes
- ▶ IaC embeds reconfiguration, adaptation, and lifecycle management
 - Declarative approaches

Terraform

- ▶ PE-K
- ▶ CRUD lifecycles
- ▶ Cloud resources / APIs

Kubernetes

- ▶ full MAPE-K
- ▶ CRUD lifecycles
- ▶ Containers

Ansible

- ▶ E-K
- ▶ Flexible lifecycles
- ▶ No composition

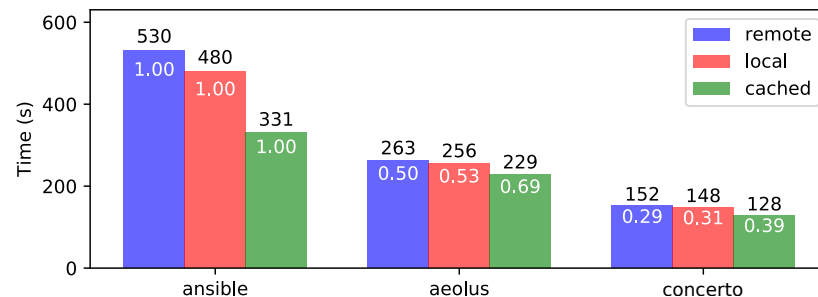
Evaluating dynamic reconfigurations

Efficiency of reconfigurations with Concerto

- ▶ **Usual metrics:** minimize downtime, time to reach the target state

Minimal OpenStack deployment

- ▶ 11 services
- ▶ 1 control node + 3 nodes



 Artifacts availability

Up to 70% gain compared to Ansible

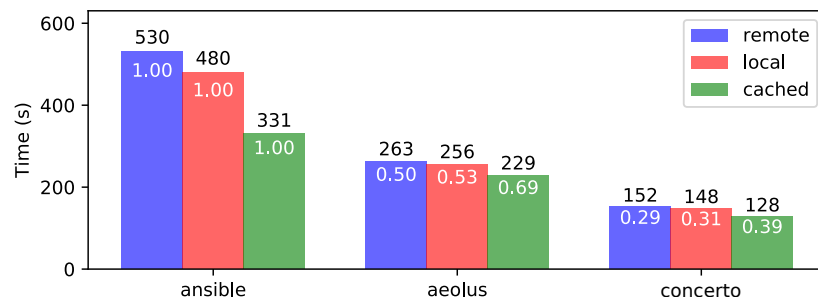
[PhD thesis of Maverick Chardet]

Efficiency of reconfigurations with Concerto

- **Usual metrics:** minimize downtime, time to reach the target state

Minimal OpenStack deployment

- 11 services
- 1 control node + 3 nodes



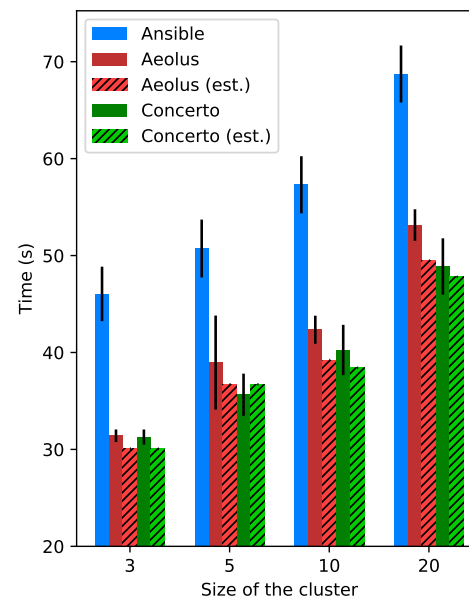
Artifacts availability

Up to 70% gain compared to Ansible

[PhD thesis of Maverick Chardet]

Reconfiguration of a multi-region OpenStack

- from single MDB to a Galera cluster
- 1 control node + n nodes for Galera



Artifacts availability

*Up to 40% gain
compared to Ansible*

Reconfigurations of a CPS



Distributed Arctic Observatory

[PhD thesis of Antoine Omond]

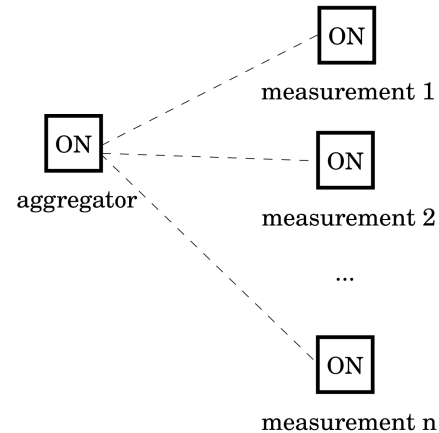
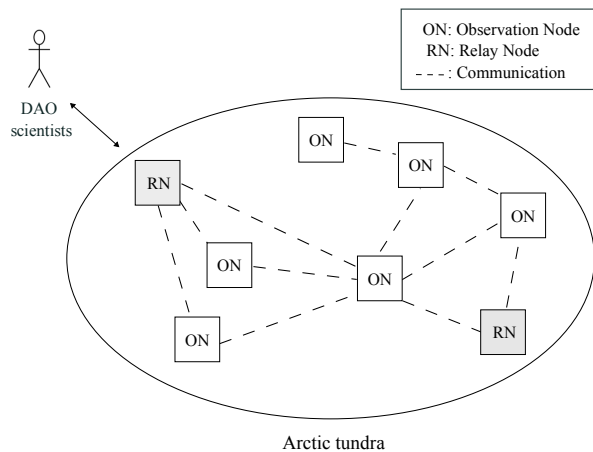


Reconfigurations of a CPS



Distributed Arctic Observatory

[PhD thesis of Antoine Omond]



- ▶ Collaborating units
- ▶ Reconfigurable units
 - Change Observations
 - Update software, etc.
- ▶ Batteries only
- ▶ Asynchronous sleeping nodes
 - Observations first

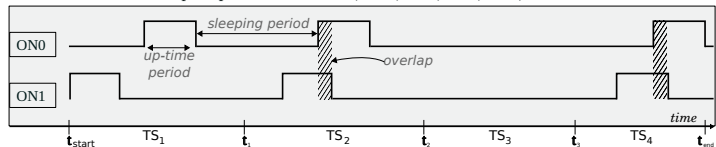
Experimental evaluation

Scenario: update collaborating units [4], [5]

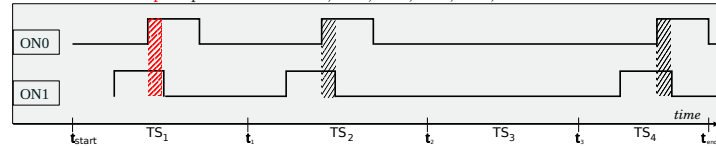
Studying trade-offs for real DAO deployment:

- Synchronizing more the units by increasing overlaps
- Leveraging relay nodes for asynchronous communications
- **Parameters:** number of overlaps, order of overlaps
- **Metrics:** reconfiguration time, energy consumption of units (including relay nodes)

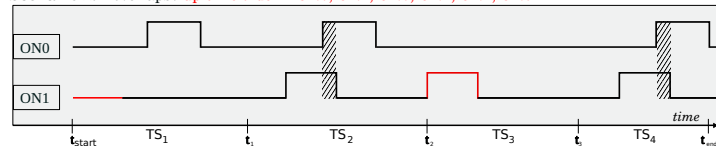
Base scenario: 2 overlaps / up-time order = ON1, ON0, ON1, ON0, ON1, ON0



Scenario 1: 3 overlaps / up-time order = ON1, ON0, ON1, ON0, ON1, ON0



Scenario 2: 2 overlaps / up-time order = ON0, ON1, ON0, ON1, ON1, ON0



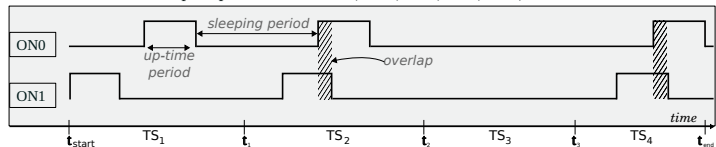
Experimental evaluation

Scenario: update collaborating units [4], [5]

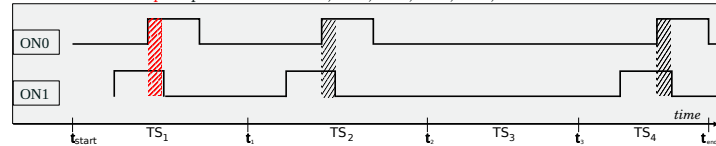
Studying trade-offs for real DAO deployment:

- Synchronizing more the units by increasing overlaps
- Leveraging relay nodes for asynchronous communications
- **Parameters:** number of overlaps, order of overlaps
- **Metrics:** reconfiguration time, energy consumption of units (including relay nodes)

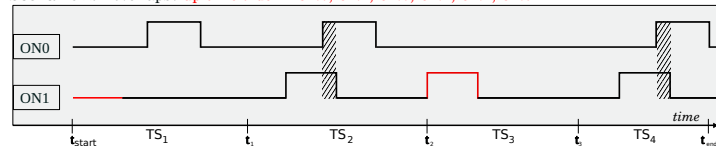
Base scenario: 2 overlaps / up-time order = ON1, ON0, ON1, ON0, ON1, ON0



Scenario 1: 3 overlaps / up-time order = ON1, ON0, ON1, ON0, ON1, ON0



Scenario 2: 2 overlaps / up-time order = ON0, ON1, ON0, ON1, ON1, ON0



- Real experiment
 - Cluster of Raspberries (Samouraï project)
 - **BUT** simulated sleeping periods
- Energy at scale studied through **simulation**
- ... no way to have real environmental conditions

⇒ *Integration of controlled environmental conditions in SLICES-FR? (GTEB résilience)*

Resilience of digital infrastructures

Critical digital infrastructures

Societies rely on digital infrastructures (ICT)

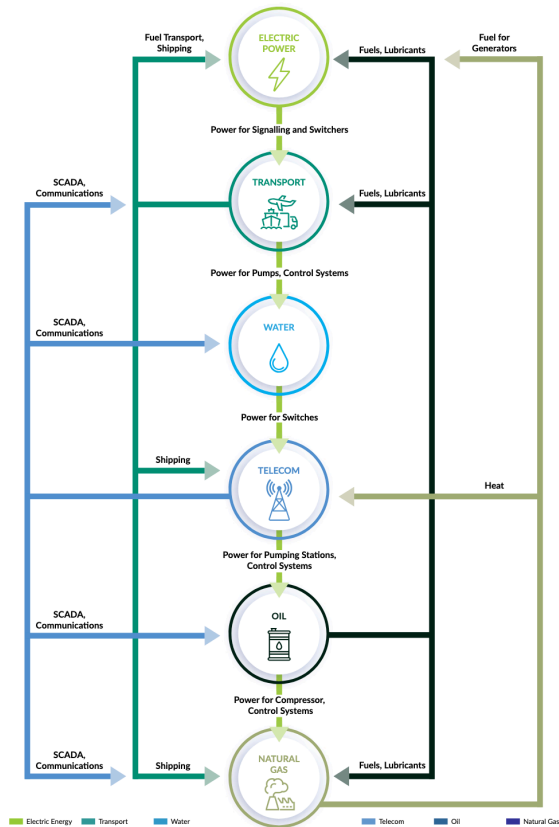


Critical digital infrastructures

Societies rely on digital infrastructures (ICT)

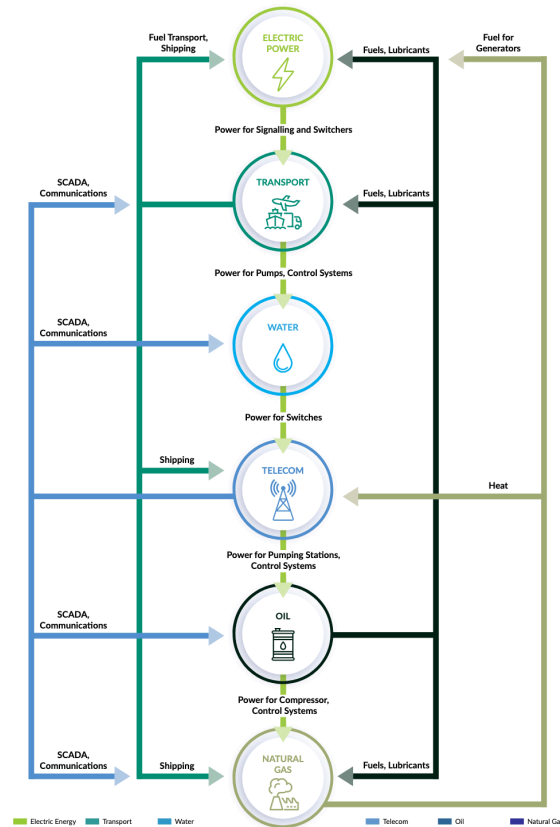


Dependencies between critical infrastructures

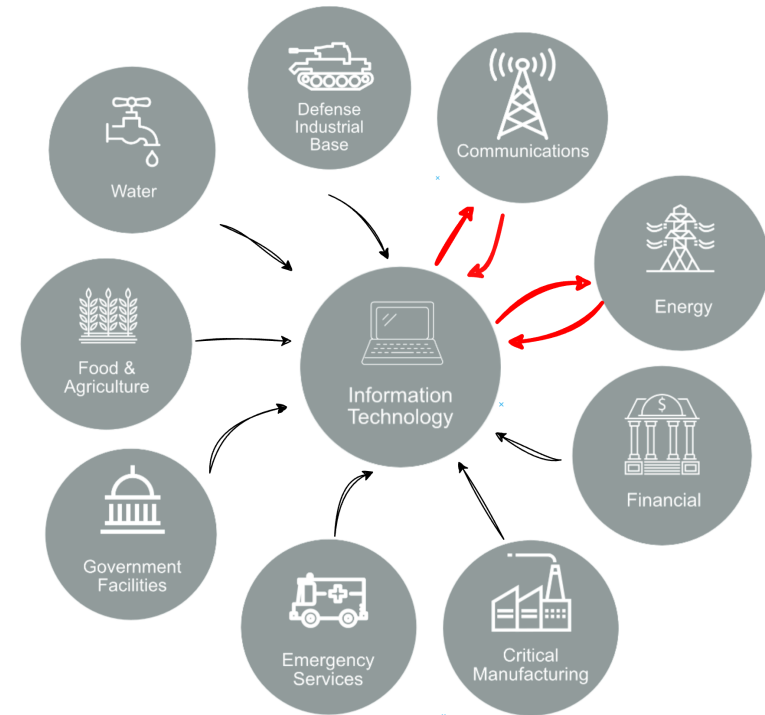


Source: National Strategy on the Resilience of Critical Entities
2026-2029

Dependencies between critical infrastructures



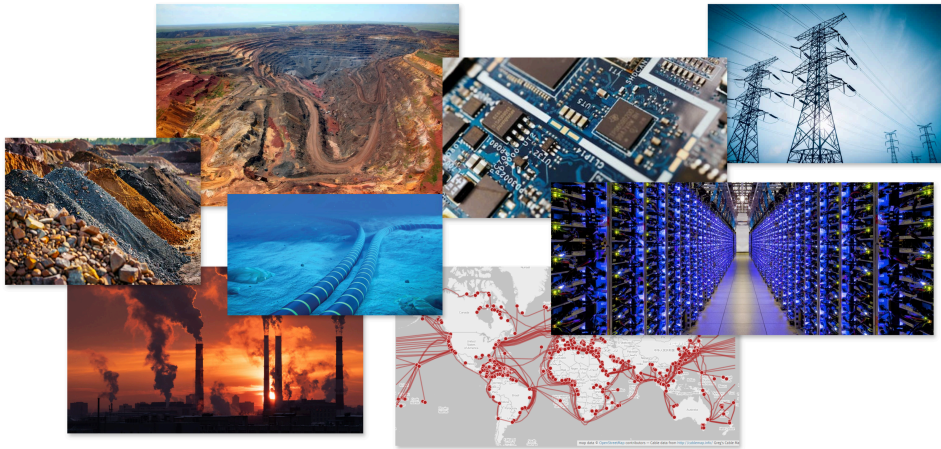
! Loops



Source: National Strategy on the Resilience of Critical Entities
2026-2029

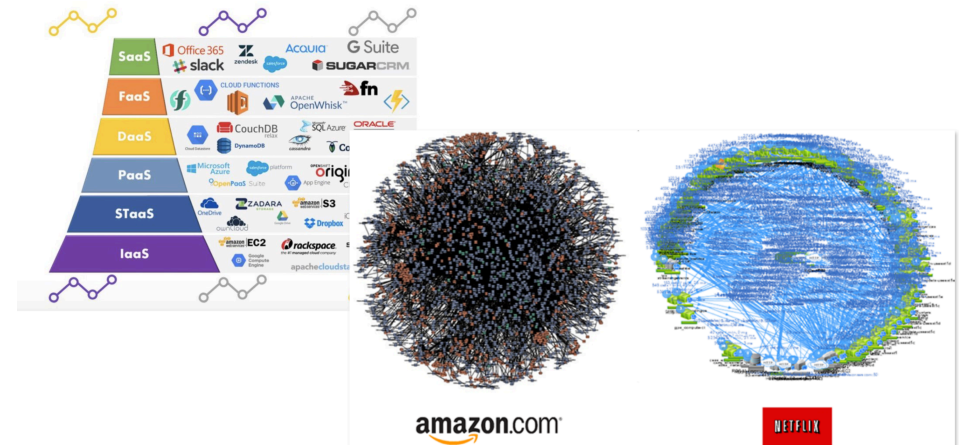
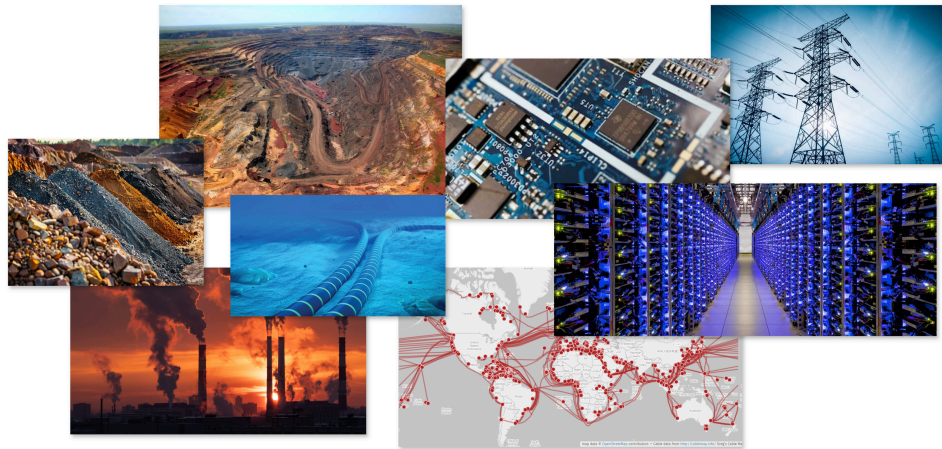
Service-Oriented Digital Infrastructure

Definition: set of hardware or software components/services required to offer higher-level services.



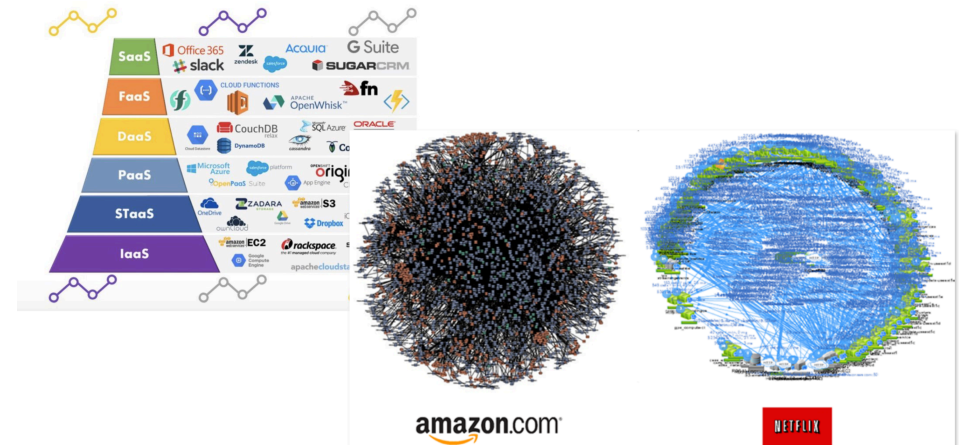
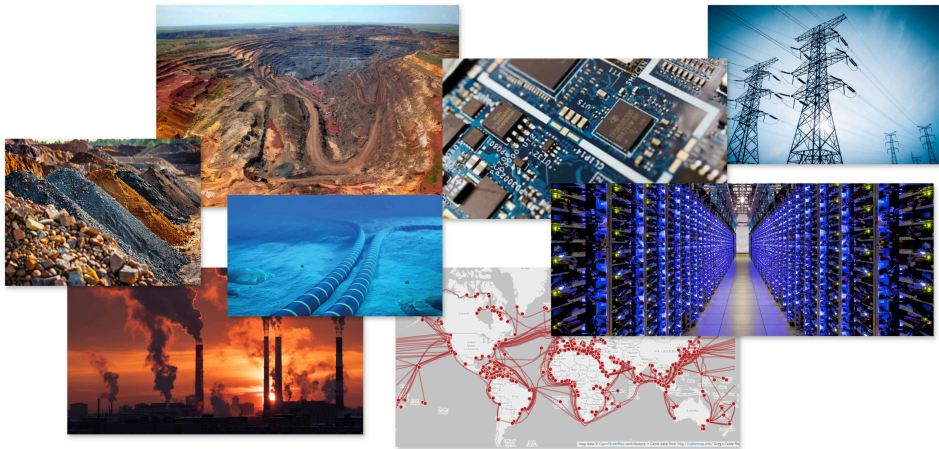
Service-Oriented Digital Infrastructure

Definition: set of hardware or software components/services required to offer higher-level services.



Service-Oriented Digital Infrastructure

Definition: set of hardware or software components/services required to offer higher-level services.



⇒ *Graph of interdependent services*

Examples: Public and private clouds, Multi-clouds, edge, continuum, on-premise infrastructures, *Cyber Physical Systems*

Fragilities

Complex evolving Systems with many dependencies

⇒ *Internal and worldwide cascading outages*

Fragilities

Complex evolving Systems with many dependencies

⇒ *Internal and worldwide cascading outages*

1. Network config
2. Undersea cable cuts
3. Incompatible metadata in CDN



Jan. 2025, 50h
Sept. 2025
Oct. 2025, , 10h

*Bad automated update
quota check system*



June 2025, 3h

*Race condition in
Amazon's DynamoDB's DNS*



Oct. 2025, 15h

*Larger than expected file
replication*



Nov. 2025, 8h

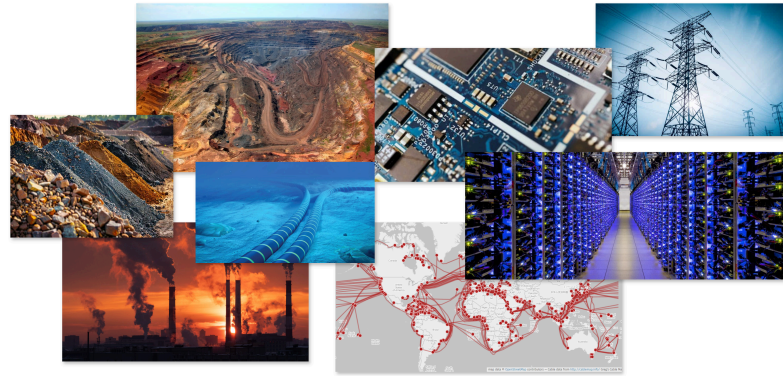
Resilience

Resilience

I use resilience as an integrative concept that encompasses system **robustness**, **survivability**, **recovery**, and **adaptability**



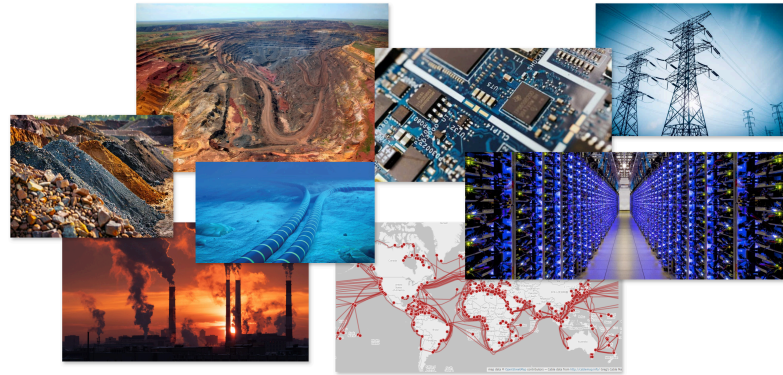
Are ICTs resilient?



*ICTs are generally **resilient** (i.e., able to recover after incidents)*

BUT they rely on the following strong **hypothesis**

Are ICTs resilient?

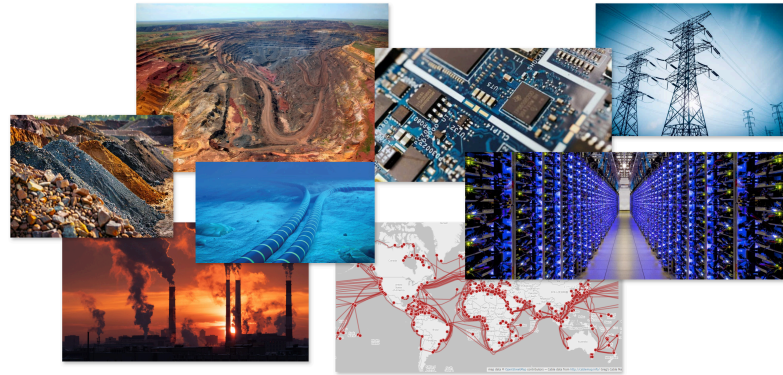


*ICTs are generally **resilient** (i.e., able to recover after incidents)*

BUT they rely on the following strong **hypothesis**

- **Rare** and **isolated** (local) problems

Are ICTs resilient?

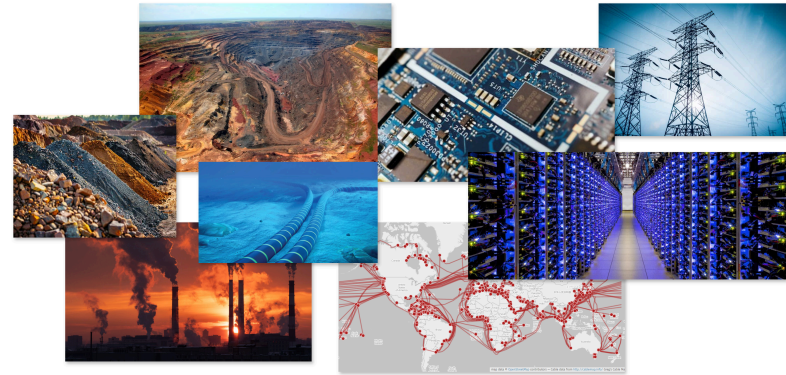


*ICTs are generally **resilient** (i.e., able to recover after incidents)*

BUT they rely on the following strong **hypothesis**

- **Rare** and **isolated** (local) problems
- **Infinite resources** (computing, storage, energy, water, batteries, etc.)

Are ICTs resilient?



*ICTs are generally **resilient** (i.e., able to recover after incidents)*

BUT they rely on the following strong **hypothesis**

- **Rare** and **isolated** (local) problems
- **Infinite resources** (computing, storage, energy, water, batteries, etc.)
- **Stability** of external services (electricity, hardware supply etc.)

Polycrises increase fragilities

Which crises?

- climate change
- natural resource scarcity
- geopolitical instability

⇒  **Hypothesis**



Fluctuation

An **irregular** rising and falling in number or amount

Examples of fluctuations

**New Texas law allows
ERCOT to disconnect
data centers during peak
events**

Examples of fluctuations

New Texas law allows

ERCO

data

even

Feature

How Agriculture and Data Centers
Compete for the Great Lakes' Most
Precious Resource

Examples of fluctuations



Examples of fluctuations



Examples of fluctuations

New Texas law allows

ERCO

data

ev

Feature

How Agriculture and Data Centers

How the April 28, 2025, power outage in Portugal and Spain impacted Internet traffic and connectivity

2025-04-28

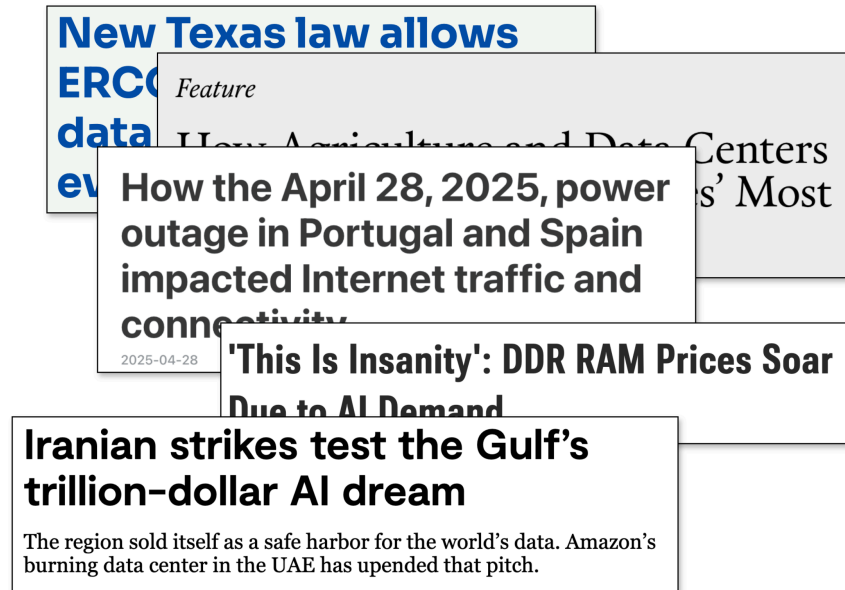
'This Is Insanity': DDR RAM Prices Soar

Due to AI Demand

Iranian strikes test the Gulf's trillion-dollar AI dream

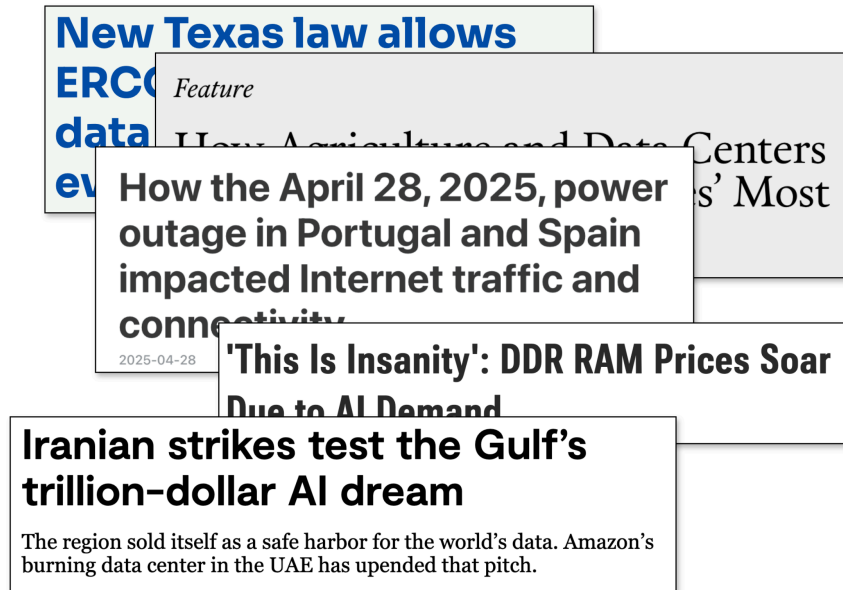
The region sold itself as a safe harbor for the world's data. Amazon's burning data center in the UAE has upended that pitch.

Examples of fluctuations



- ▶ Fluctuations in access to water
- ▶ Temperature fluctuations
- ▶ Electrical fluctuations
- ▶ Fluctuations on hardware supply
- ▶ Fluctuation in network quality or connectivity
- ▶ Fluctuations to access data, software, services, applications
- ▶ etc.

Examples of fluctuations



- ▶ Fluctuations in access to water
- ▶ Temperature fluctuations
- ▶ Electrical fluctuations
- ▶ Fluctuations on hardware supply
- ▶ Fluctuation in network quality or connectivity
- ▶ Fluctuations to access data, software, services, applications
- ▶ etc.

*“Modern Internet services are composed of hundreds of interdependent systems spanning dozens of geo-distributed data centers. At this scale, seemingly **rare natural disasters, such as hurricanes blowing down power lines and flooding, occur regularly**”*

— Facebook researchers, OSDI 2018

New Inria team - *RIFT*

Resilience of digital Infrastructures to resource FlucTuations 🐙

⚠️ Still under creation

⇒ Following the current *STACK* research group of Adrien Lebre



Daniel Balouek



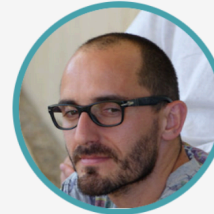
Carlos Gonzalez



Baptiste Jonglez



Remous Aris Koutsiamanis



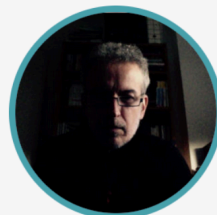
Adrien Lebre



Thomas Ledoux



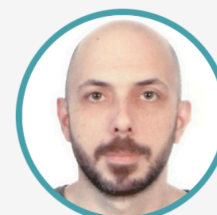
Jean-Marc Menaud



Jacques Noyé



Kandaraj Piamrat



Guillaume Rosinosky



Mario Südholt

Research program

- ▶ *Axis 1*: Considering limitations and degradations
 - Conectivity, energy, hardware

Research program

- ▶ *Axis 1*: Considering limitations and degradations
 - Connectivity, energy, hardware
- ▶ *Axis 2*: Handling the temporal scales of fluctuations
 - urgency, high frequency, long-term permanent

Research program

- ▶ **Axis 1:** Considering limitations and degradations
 - Connectivity, energy, hardware
- ▶ **Axis 2:** Handling the temporal scales of fluctuations
 - urgency, high frequency, long-term permanent
- ▶ **Axis 3:** Handling the spatial scales of fluctuations
 - geography and territoriality, sovereignty

Research program

- ▶ **Axis 1:** Considering limitations and degradations
 - Connectivity, energy, hardware
- ▶ **Axis 2:** Handling the temporal scales of fluctuations
 - urgency, high frequency, long-term permanent
- ▶ **Axis 3:** Handling the spatial scales of fluctuations
 - geography and territoriality, sovereignty
- ▶ **Axis 4** Evaluating resiliency
 - Metrics, benchmarks, resilience-oriented platforms

Evaluating resilience?

Overall idea

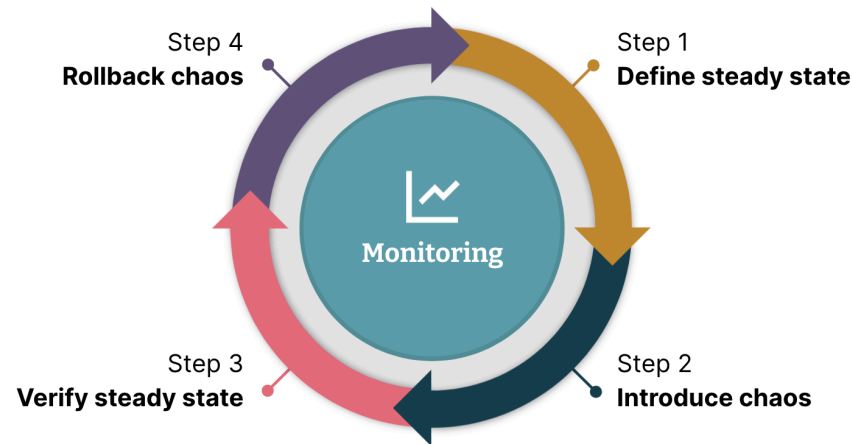
**Being able to build controlled abnormal and reproducible conditions
for systems and networks!**

— — —

Chaos testing

- ▶ 2008 - Born at Netflix
- ▶ 2010 - Chaos Monkey
- ▶ 2016 - Gremlin
- ▶ 2019 - Chaos Mesh, Litmus (CNCF)

⇒ *Integration of Chaos in SLICES-FR? (GTEB résilience)* [Internship Martin Plaud]



Resilience of Kubernetes [6], [7] (or IaC in general)

TABLE I: Comparison of Kubernetes Fault Injection Tools and Frameworks

Tool/Framework	Pod Faults	Net Faults	Ctrl Plane	Load Gen.	App Metrics	Cloud Edge
Chaosmesh [38]	✓	✓	×	×	×	✓
Litmuschaos [39]	✓	✓	×	×	×	✓
Gremlin [23]	✓	✓	×	×	✓	×
Chaosblade [40]	✓	✓	×	×	×	✓
Chaostoolkit [41]	✓	✓	×	×	×	✓
Powerfulseal [42]	✓	✓	×	×	×	✓
chaoskube [43]	✓	×	×	×	×	✓
Mutiny [8]	✓	✓	✓	×	×	×
Our Work	✓	✓	✓	✓	✓	✓

POD FAULTS: Pod/Node level fault injection.

NET FAULTS: Network fault injection capability.

CTRL PLANE: Control-plane fault injection.

LOAD GEN.: Workload generation integration.

APP METRICS: Application performance metrics.

CLOUD-EDGE: Cloud-edge coordination support.

Resilience Evaluation of Kubernetes in Cloud-Edge Environments via Failure Injection

Zihao Chen¹, Mohammad Goudarzi¹, Adel Nadjaran Toosi²

¹The Faculty of Information Technology, Monash University, Australia

²School of Computing and Information Systems, The University of Melbourne, Australia

Resilience of Kubernetes [6], [7] (or IaC in general)

TABLE I: Comparison of Kubernetes Fault Injection Tools and Frameworks

Tool/Framework	Pod Faults	Net Faults	Ctrl Plane	Load Gen.	App Metrics	Cloud Edge
Chaosmesh [38]	✓	✓	×	×	×	✓
Litmuschaos [39]	✓	✓	×	×	×	✓
Gremlin [23]	✓	✓	×	×	✓	×
Chaosblade [40]	✓	✓	×	×	×	✓
Chaostoolkit [41]	✓	✓	×	×	×	✓
Powerfulseal [42]	✓	✓	×	×	×	✓
chaoskube [43]	✓	×	×	×	×	✓
Mutiny [8]	✓	✓	✓	×	×	×
Our Work	✓	✓	✓	✓	✓	✓

POD FAULTS: Pod/Node level fault injection.

NET FAULTS: Network fault injection capability.

CTRL PLANE: Control-plane fault injection.

LOAD GEN.: Workload generation integration.

APP METRICS: Application performance metrics.

CLOUD-EDGE: Cloud-edge coordination support.

Resilience Evaluation of Kubernetes in Cloud-Edge Environments via Failure Injection

Zihao Chen¹, Mohammad Goudarzi¹, Adel Nadjaran Toosi²

¹The Faculty of Information Technology, Monash University, Australia

²School of Computing and Information Systems, The University of Melbourne, Australia

Mutiny! How does Kubernetes fail, and what can we do about it?

Marco Barletta*, Marcello Cinque*, Catello Di Martino[†], Zbigniew T. Kalbarczyk[‡], and Ravishankar K. Iyer[‡]

*Università degli Studi di Napoli Federico II, Naples, NA 80125, Italy

[†]Nokia Bell Labs, Sao Paulo, SP 05069-010, Brazil

[‡]University of Illinois at Urbana-Champaign, Urbana-Champaign, IL 61801, USA

[PhDs of Nathan Fouéré (PEPR Cloud) and Lucas Deloison (PEPR NumEco) -> starting in October]

Extending chaos to other crises

Chaos's tools are **extensive**:

- latency, bandwidth, pod, DNS, HTTP, platforms, etc.

Extending chaos to other crises

Chaos's tools are **extensive**:

- latency, bandwidth, pod, DNS, HTTP, platforms, etc.

⇒ ***But limited to faults or degradations***

Other kind of issues related to crises:

- High frequency of issues
- Urgency of issues
- Configuration drifts
- Degradations of non-IT services (electricity, water, etc.)
- Dynamic limitations (energy, resources, costs, etc.)
- Dynamic introduction of heterogeneous hardware, etc.

Time limitations

Urgency

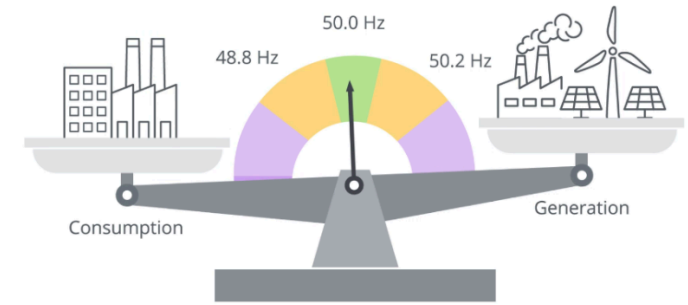


Electrical constraints

*PhDs of Nathan Rabier (PEPR
Cloud)*



Le réseau
de transport
d'électricité



Source: regelleistung.net

Time limitations

Urgency

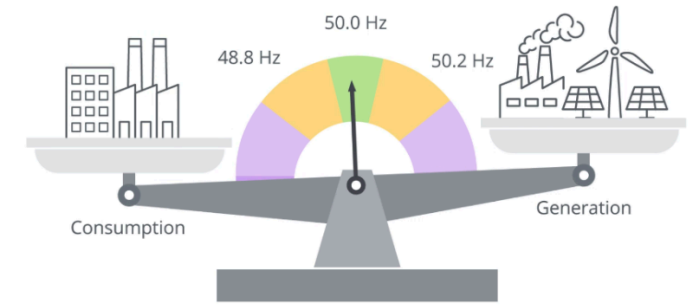


Electrical constraints

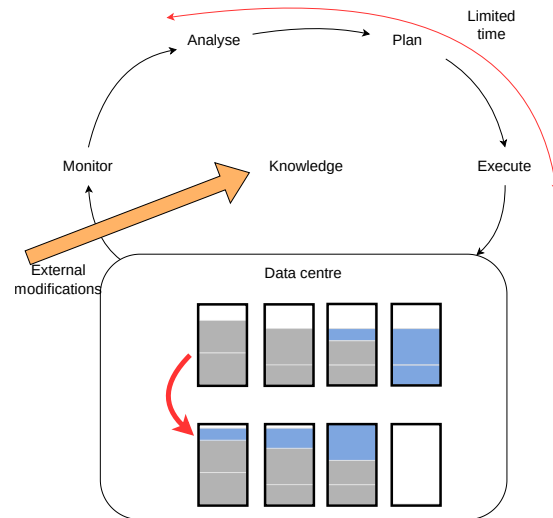
PhDs of Nathan Rabier (PEPR Cloud)



Le réseau
de transport
d'électricité



Source: regelleistung.net



Time limitations

Urgency

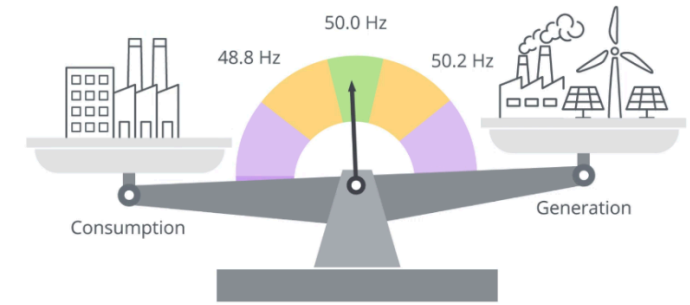


Electrical constraints

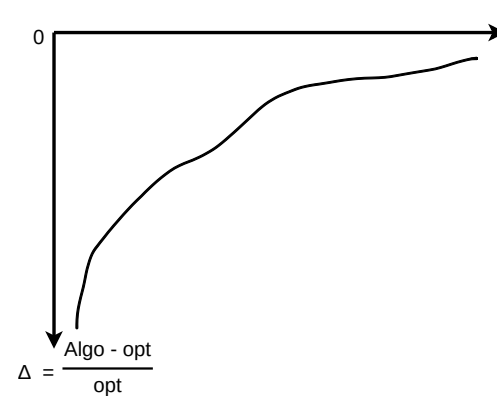
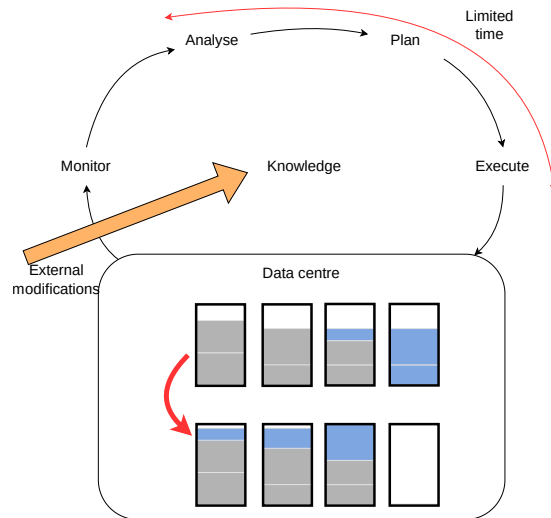
PhDs of Nathan Rabier (PEPR Cloud)



Le réseau
de transport
d'électricité



Source: regelleistung.net

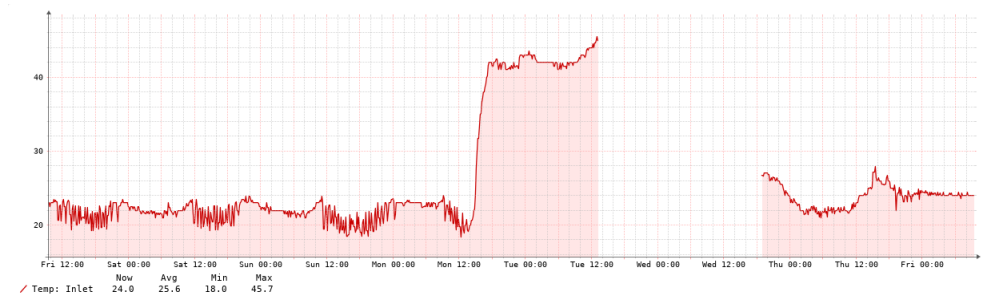


- ▶ Anytime properties of existing techniques[8]
- ▶ Real experiments will be conducted on the full MAPE-K with Kubernetes

Exposure to heat waves, cold snap, humidity, etc.



Grid'5000 Rennes

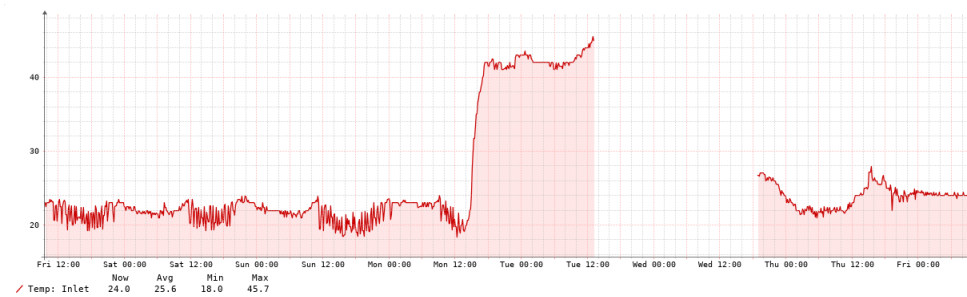


[Blog post from Deuxfleurs](#)

Exposure to heat waves, cold snap, humidity, etc.



Grid'5000 Rennes



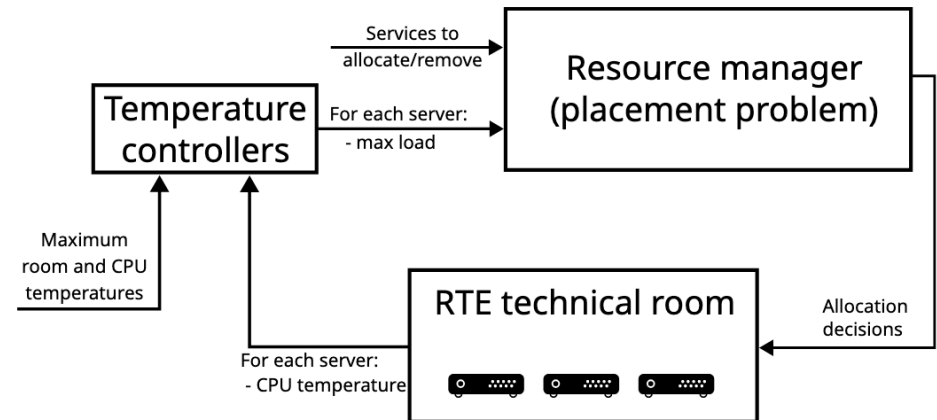
Blog post from Deuxfleurs



PEPR TASE, TASTING project

[Postdoc of Clément Mommessin]

- ▶ RTE targets to virtualize their control/command functions in electrical control station
- ▶ Buildings without insulation and too numerous to consider mass cooling
- ▶ Rodents 🐭, heat waves 🔥, cold snaps ❄️



⇒ *Integration of controlled environmental conditions in SLICES-FR?*

Survivability of services



Collectif des Hébergeurs Alternatifs, Transparents, Ouverts, Neutres et Solidaires

- Study survivability of small scale jointly managed ICTs
- Build a high level tool to evaluate survivability

Simulation-based $\Rightarrow$ ***Validate models on real infrastructures***

- Build a clone of existing infrastructures on SLICES-FR?

[PhDs of Lucien Astié]

Thank you!

GTEB SLICES-FR fragility and resilience

Help us break everything! 🪓💣🔪

- ▶ Mariam Ahhhtouche, research engineer, Inria Lyon, CITI
- ▶ Thierry Arrabal, associate professor, University of Lorraine, LORIA
- ▶ **Hélène Coullon**, associate professor HDR, IMT Atlantique, LS2N
- ▶ Didier Donsez, professor, Grenoble INP—UGA, IMAG
- ▶ Baptiste Jonglez, research engineer, Inria Rennes, LS2N
- ▶ **Remous Aris Koutsiamanis**, associate professor, IMT Atlantique, LS2N
- ▶ Laurent Lefèvre, researcher, Inria Lyon, LIP
- ▶ Adrien Luxey-Bitri, associate professor, University of Lille, CRISTAL
- ▶ Martin Plaud, apprenticeship, Inria Rennes, LS2N
- ▶ Olivier Richard, associate professor, University Grenoble Alpes, LIG
- ▶ Romain Rouvoy, professor, University of Lille, CRISTAL
- ▶ Matthieu Simonin, research engineer, Inria Rennes, IRISA
- ▶ Fabrice Valois, professor, INSA Lyon, CITI



Bibliography

Bibliography

- [1] H. Coullon, L. Henrio, F. Loulergue, and S. Robillard, “Component-based Distributed Software Reconfiguration:A Verification-oriented Survey,” *ACM Comput. Surv.*, vol. 56, no. 1, Aug. 2023, doi: [10.1145/3595376](https://doi.org/10.1145/3595376).
- [2] J. O. Kephart and D. M. Chess, “The Vision of Autonomic Computing,” *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003, doi: [10.1109/MC.2003.1160055](https://doi.org/10.1109/MC.2003.1160055).
- [3] M. Chardet, H. Coullon, and S. Robillard, “Toward safe and efficient reconfiguration with Concerto,” *Science of Computer Programming*, vol. 203, p. 102582, 2021, doi: <https://doi.org/10.1016/j.scico.2020.102582>.
- [4] A. Omond, H. Coullon, I. Rais, and O. Anshus, “Leveraging Relay Nodes to Deploy and Update Services in a CPS with Sleeping Nodes,” in *2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 2023, pp. 532–539. doi: [10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics60724.2023.00102](https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics60724.2023.00102).
- [5] A. Omond, I. Raïs, and H. Coullon, “Evaluating the energy consumption of adaptation tasks for a CPS in the Arctic Tundra,” in *Proceeding of the 2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, Danzhou, China: IEEE, Dec. 2023, pp. 1–8. doi: [10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics60724.2023.00122](https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics60724.2023.00122).
- [6] Z. Chen, M. Goudarzi, and A. N. Toosi, “Resilience Evaluation of Kubernetes in Cloud-Edge Environments via Failure Injection.” [Online]. Available: <https://arxiv.org/abs/2507.16109>
- [7] M. Barletta, M. Cinque, C. Di Martino, Z. T. Kalbarczyk, and R. K. Iyer, “Mutiny! How Does Kubernetes Fail, and What Can We Do About It?,” in *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE, Jun. 2024, pp. 1–14. doi: [10.1109/dsn58291.2024.00016](https://doi.org/10.1109/dsn58291.2024.00016).
- [8] S. Zilberstein, “Using Anytime Algorithms in Intelligent Systems,” *AI Mag.*, vol. 17, no. 3, pp. 73–83, Sep. 1996, doi: [10.1609/aimag.v17i3.1232](https://doi.org/10.1609/aimag.v17i3.1232).