FabOS: Hooking up Container Platforms with Time-Sensitive Networks

Manuel Eppler, Jochen Schenk, Tobias Gruner, Andreas Zirkler, Harald Müller, Andreas Blenk
Siemens AG, Munich, Germany

SIEMENS

Challenging demands on Future Factories

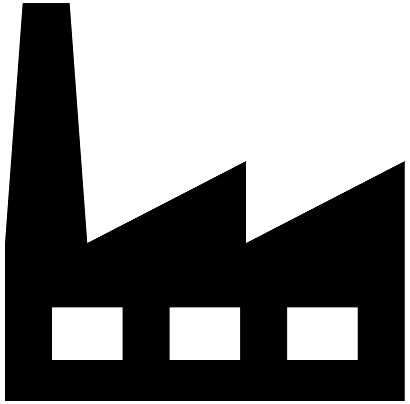
Flexible networking landscape

→ Adopt to changing Quality-of-Service requirements

Adoptive production

Frequent / daily floorplan adoptions

→ Changing setup of machines and factory OS



Real-time networking

Sustainability and compute efficiency

Flexible use of common of-the-shelf computing hardware

→ Virtual Programming Logic Controllers

Security

Security zones with real-time capabilities

→ Data-link layer access

SIEMENS

Do not reinvent the wheel!

kubernetes

- Cloud Container platform
- Flexible and efficient use of compute resources
- Easy container deployment

could become

→ ~~Is~~ the perfect compute environment for industry

Problems:

→ Container placement not aware of network requirements

→ No real-time network access

SIEMENS

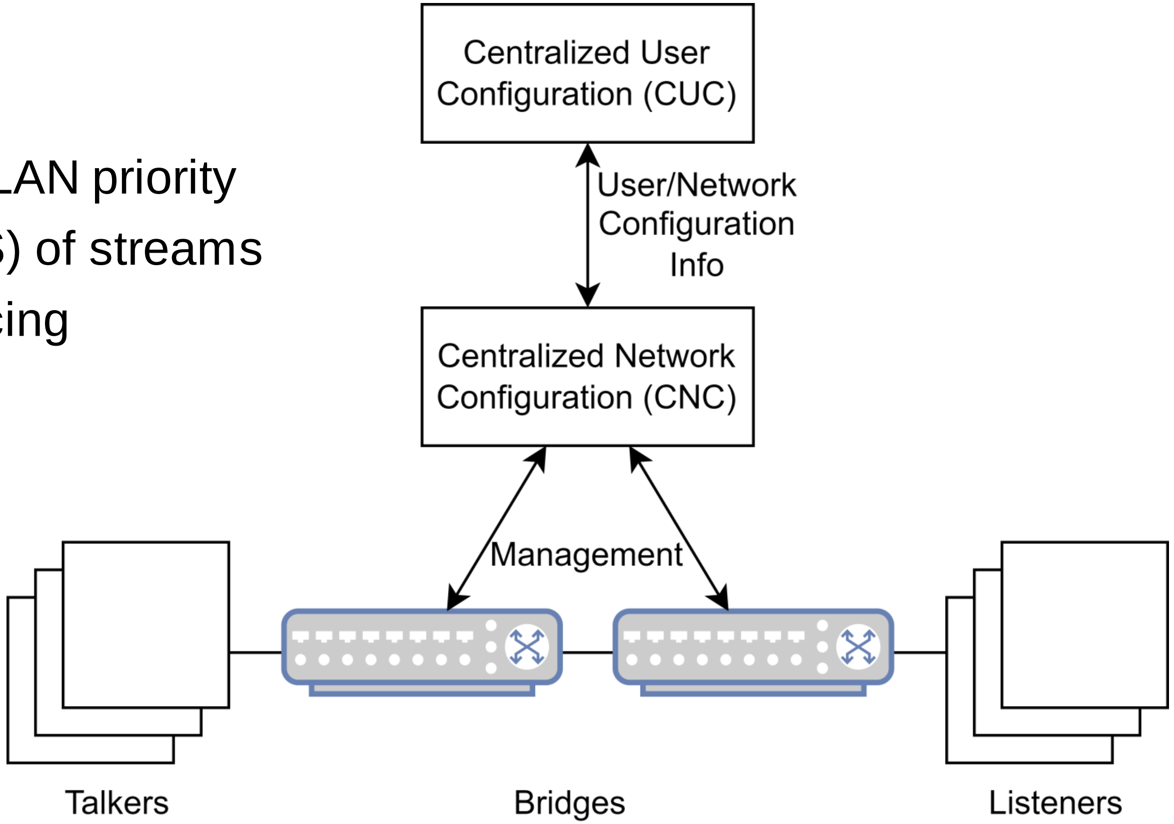
Time-Sensitive Networking (TSN)

- Promising solution for industrial networks
- Flexible network usage
- Stream detection via MAC address, VLAN ID, and VLAN priority
- Mechanisms for Guaranteed Quality-of-Service (QoS) of streams
  - E.g. Strict priority with per-stream filtering and policing
- Fully centralized model

Problem:

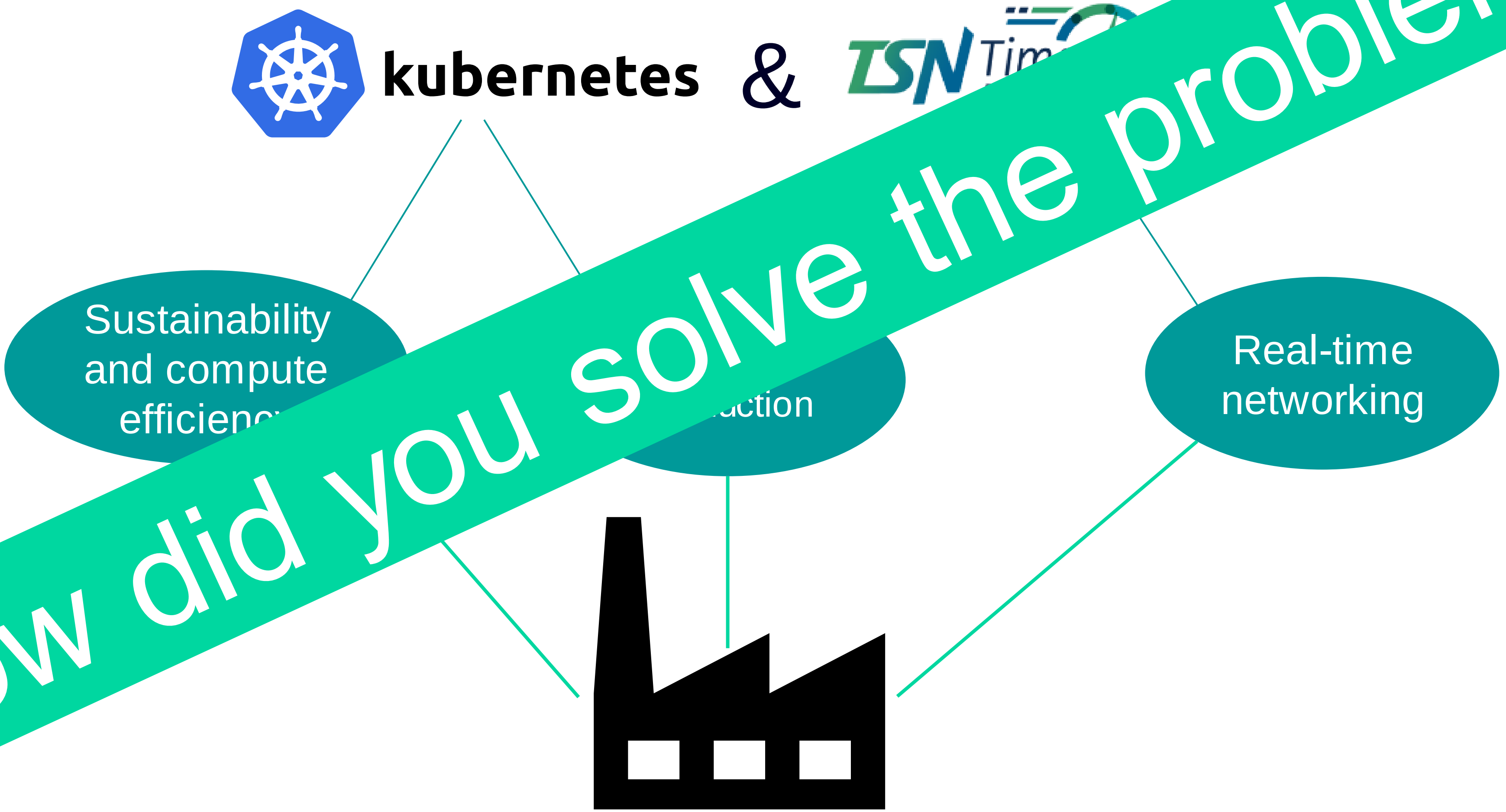
→ No implementation of CUC & CNC

→ No interface to Kubernetes



IEEE 802.1Qcc Fully centralized network model

The FabOS Architecture:
Hooking up Container Platforms with Time-Sensitive Networks

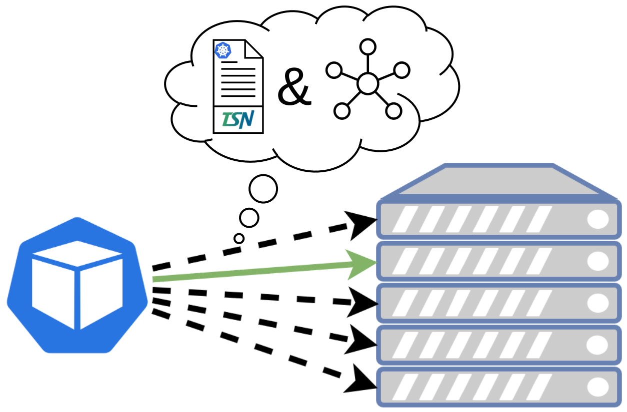


How did you solve the problems?

SIEMENS

The proposed Architecture in 3 Steps

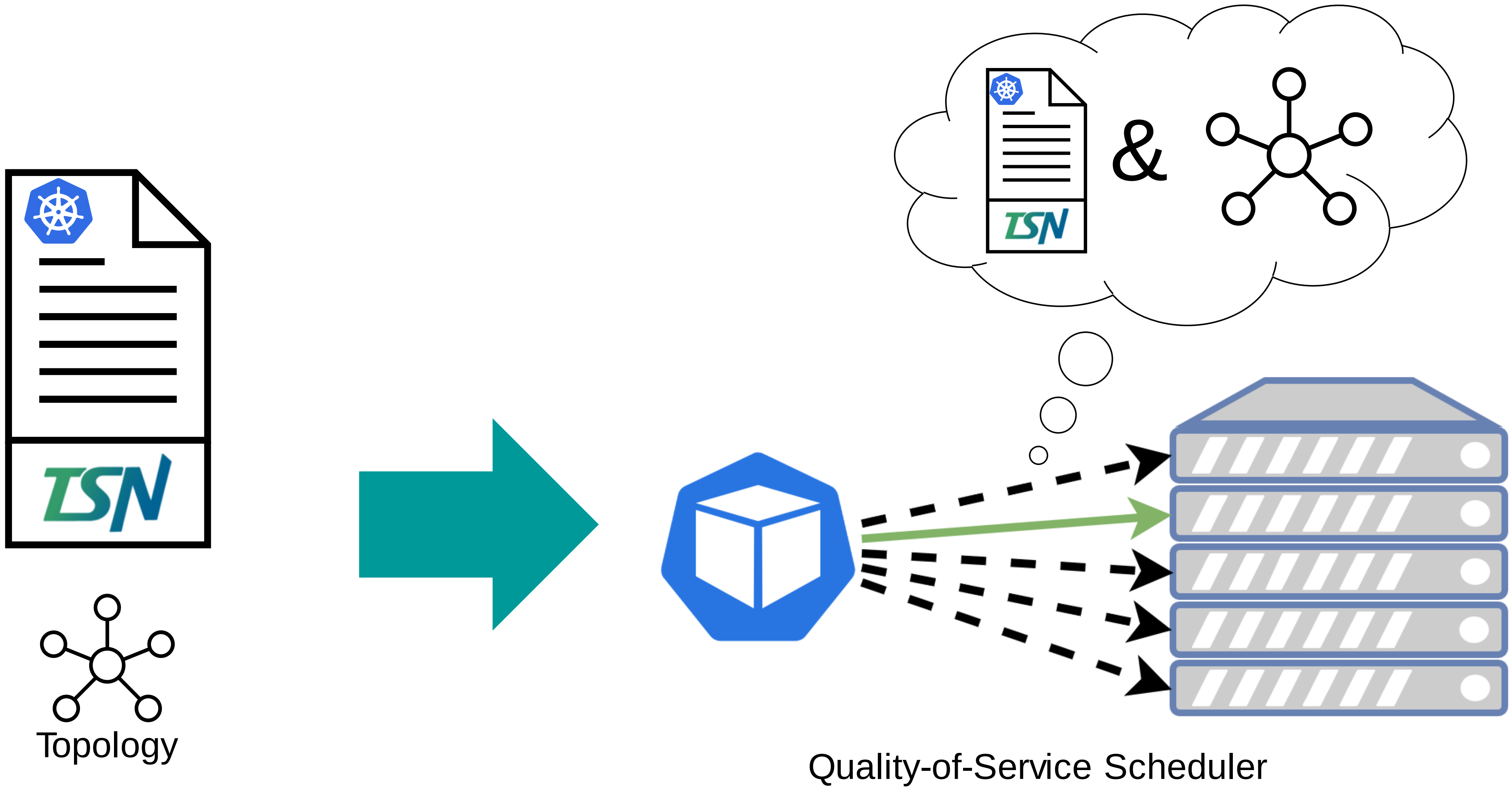
Step 1: Network aware Container Placement



Step 2: TSN Network Configuration

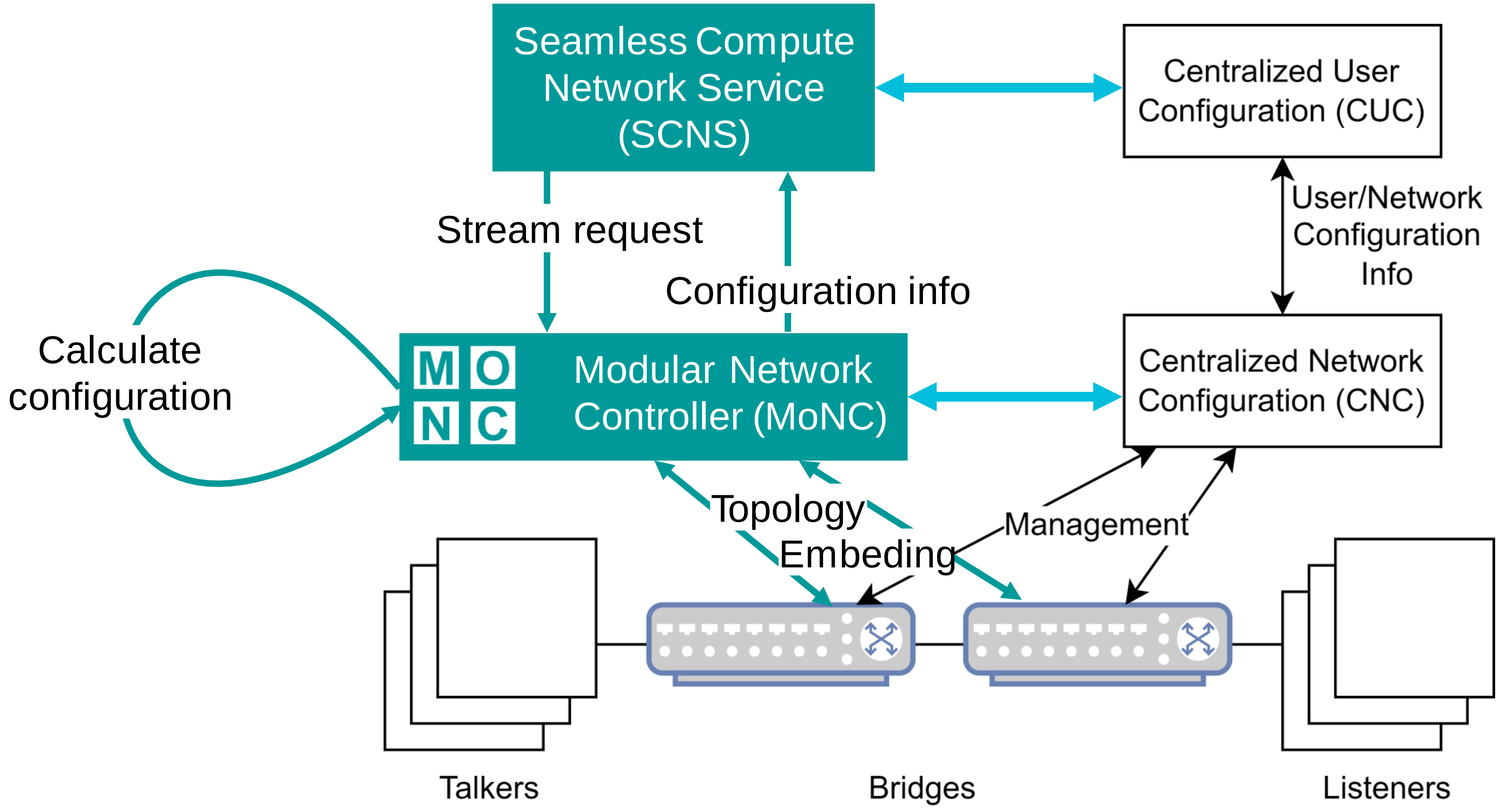
Step 3: Frame Tagging for TSN

Step 1: Network aware Container Placement



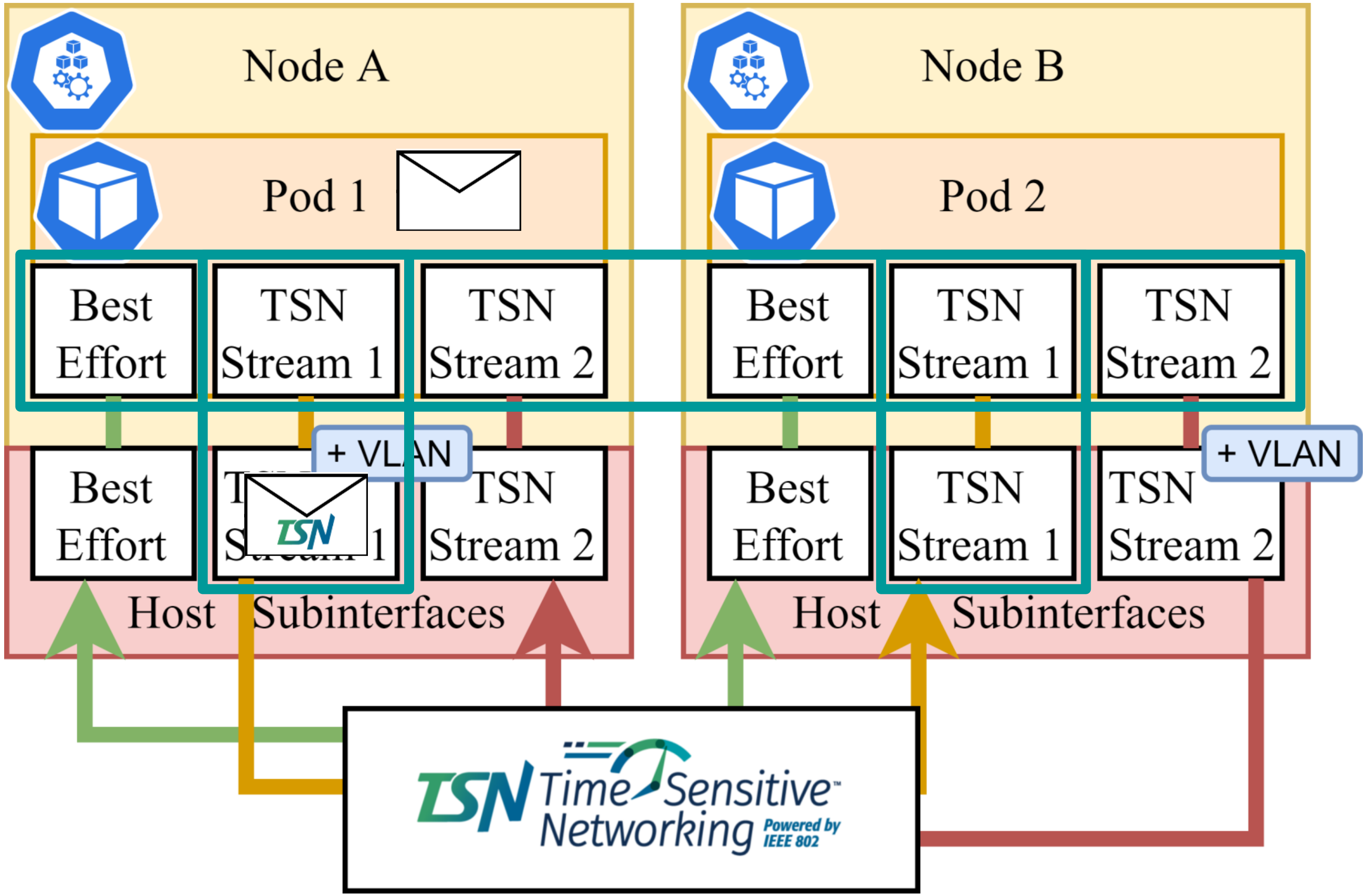
SIEMENS

Step 2: TSN Network Configuration



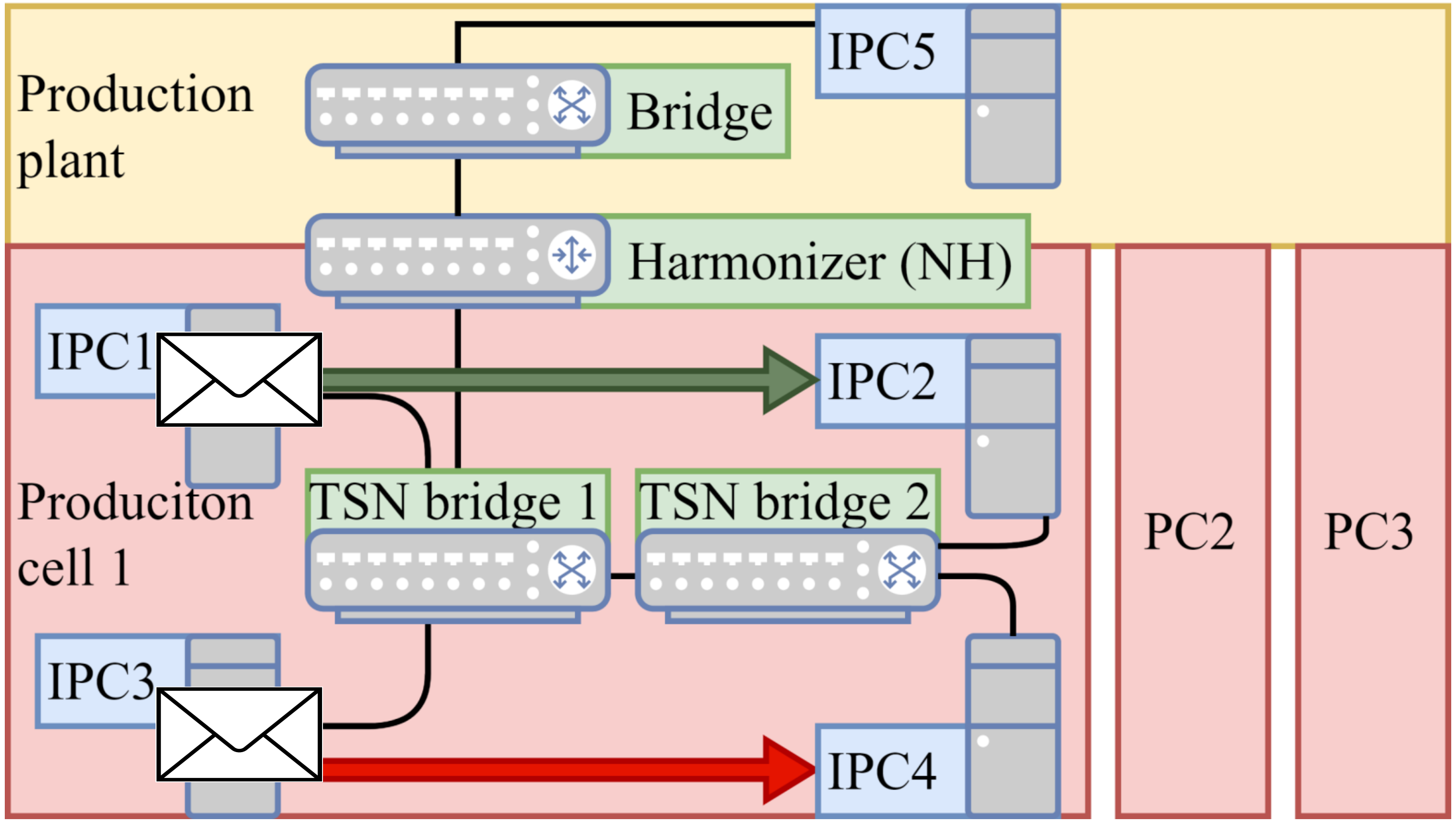
SIEMENS

Step 3: Frame Tagging for TSN

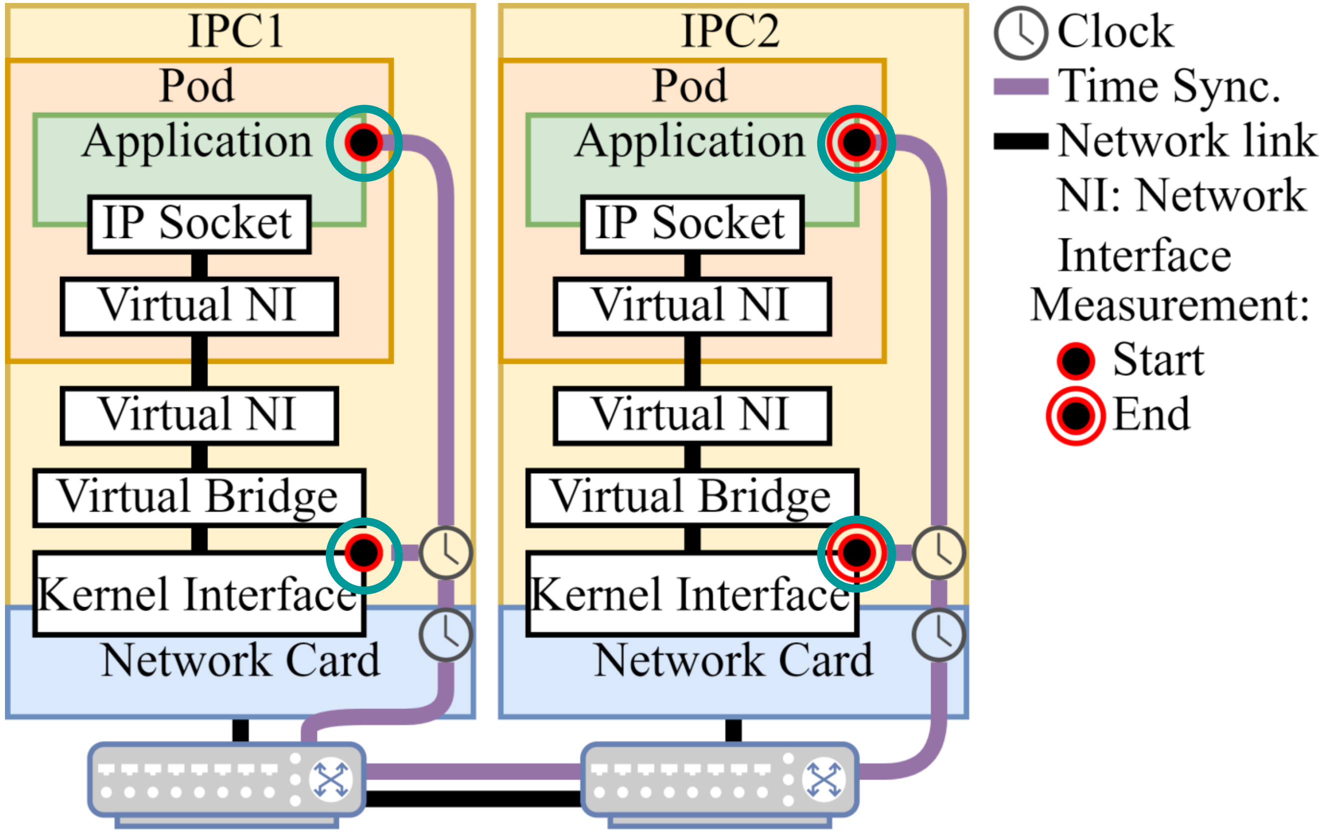


SIEMENS

Proof-of-concept implementation

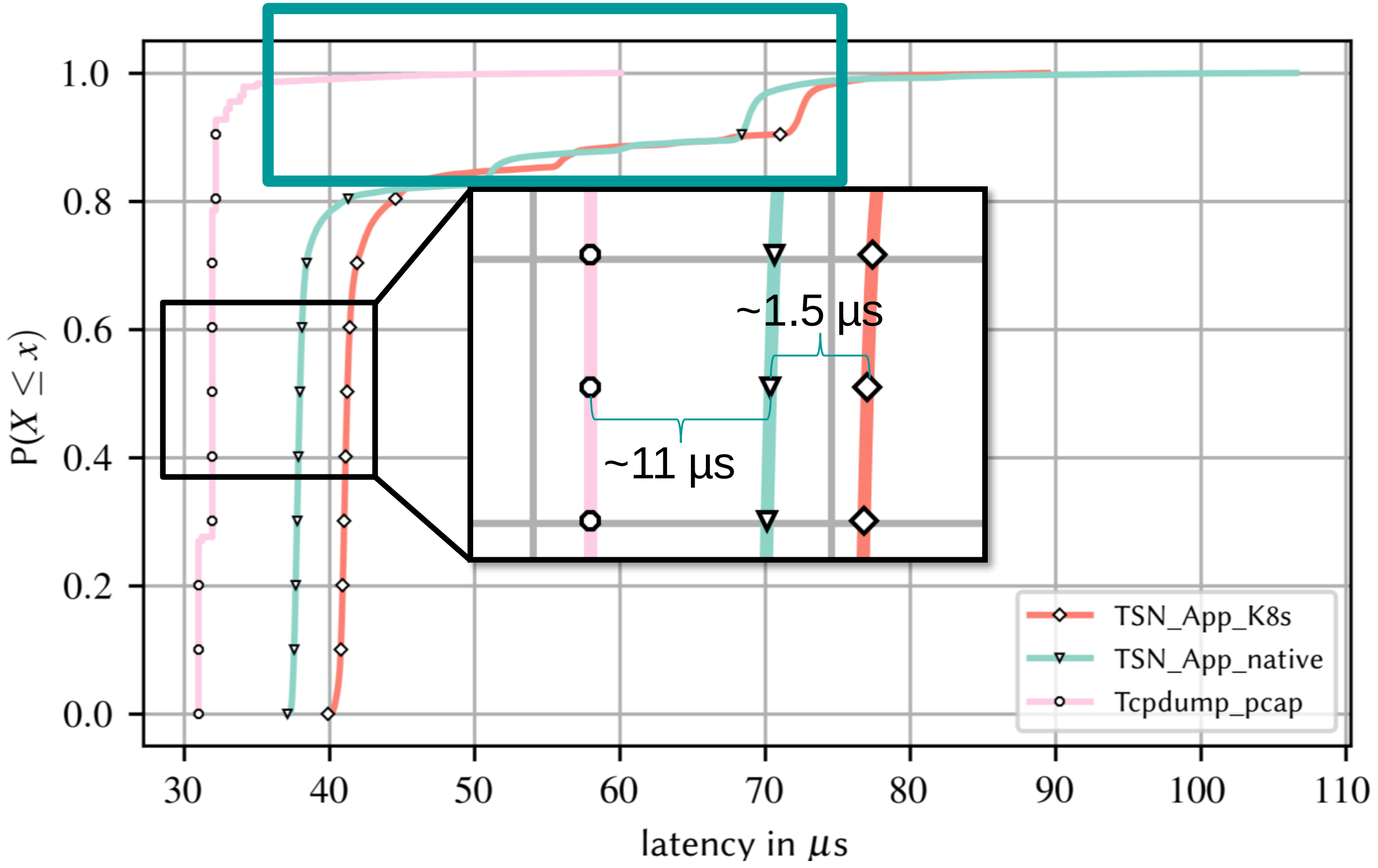


Evaluation – 3 Measurement setups



SIEMENS

Evaluation – Virtualization Overhead



Latency overhead by application:

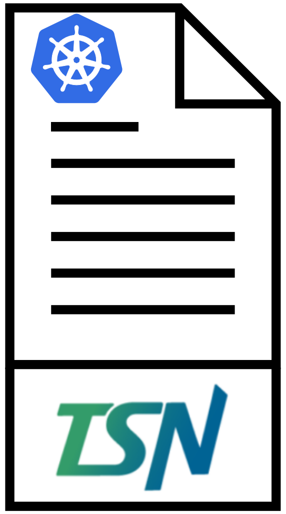
- ~33 µs Network (Linux to Linux)
- ~11 µs TSN Test App
- ~1.5 µs Kubernetes and Plugin

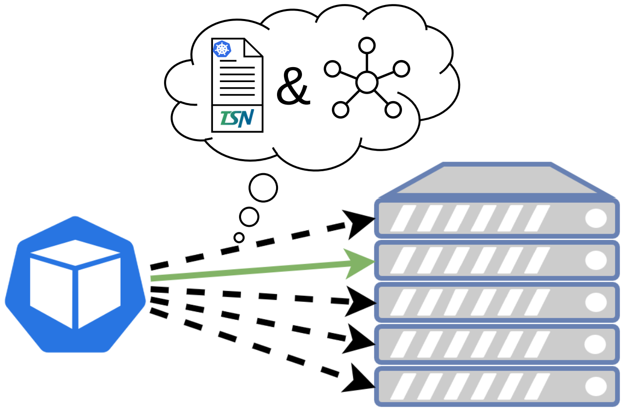
But:

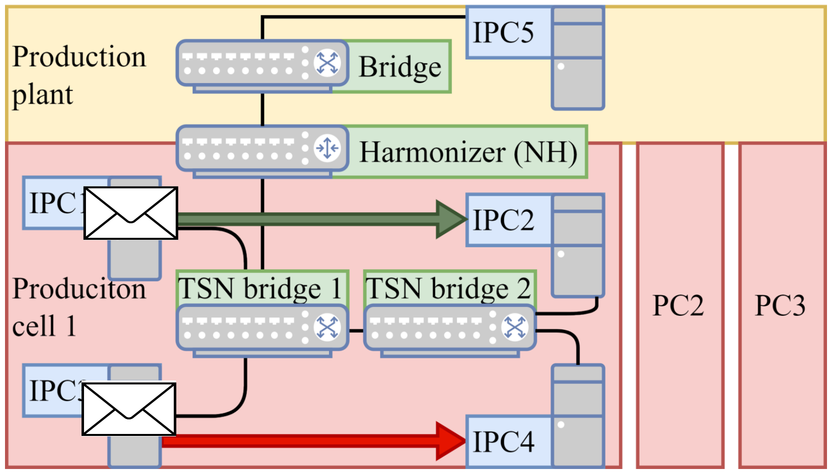
- High Latency for over 10 % of Frames

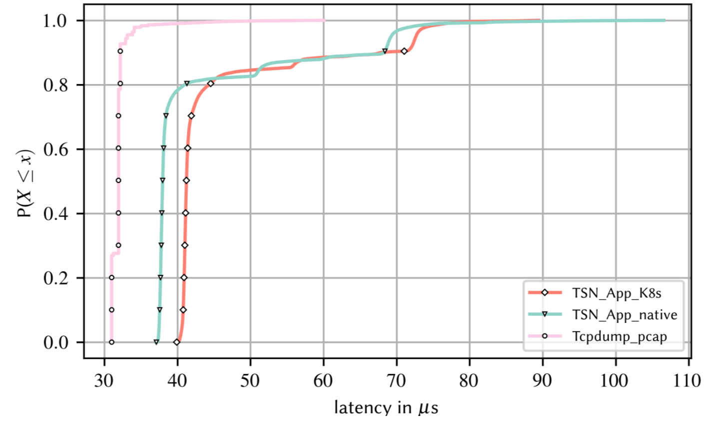
SIEMENS

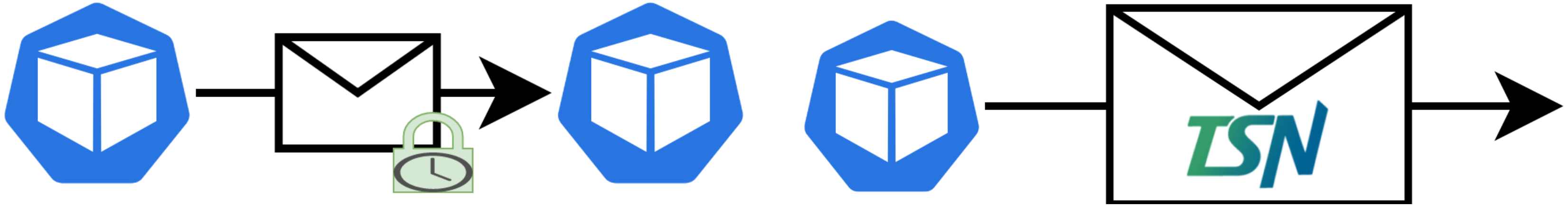
Conclusion











SIEMENS

Thank you !

Acknowledgement

This research and development project is funded by the **German Federal Ministry of Economics and Climate Action (BMWK)** within the funding measure "5G campus-networks" under the funding code 01MC22001B / 01MC22002C and supervised by the DLR Projektträger | Division Society, Innovation, Technology at the German Aerospace Center. We thank for the opportunity to work on this project.

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag

SIEMENS

Contact

Published by Siemens 2023

Manuel Eppler
Research Scientist / PhD Student
Industrial Networks and Wireless / Germany / T CED
Otto-Hahn-Ring 8
81739 Munich
Germany

Mobile +49 162 88 81 40 2

E-mail manuel.eppler@siemens.com

SIEMENS

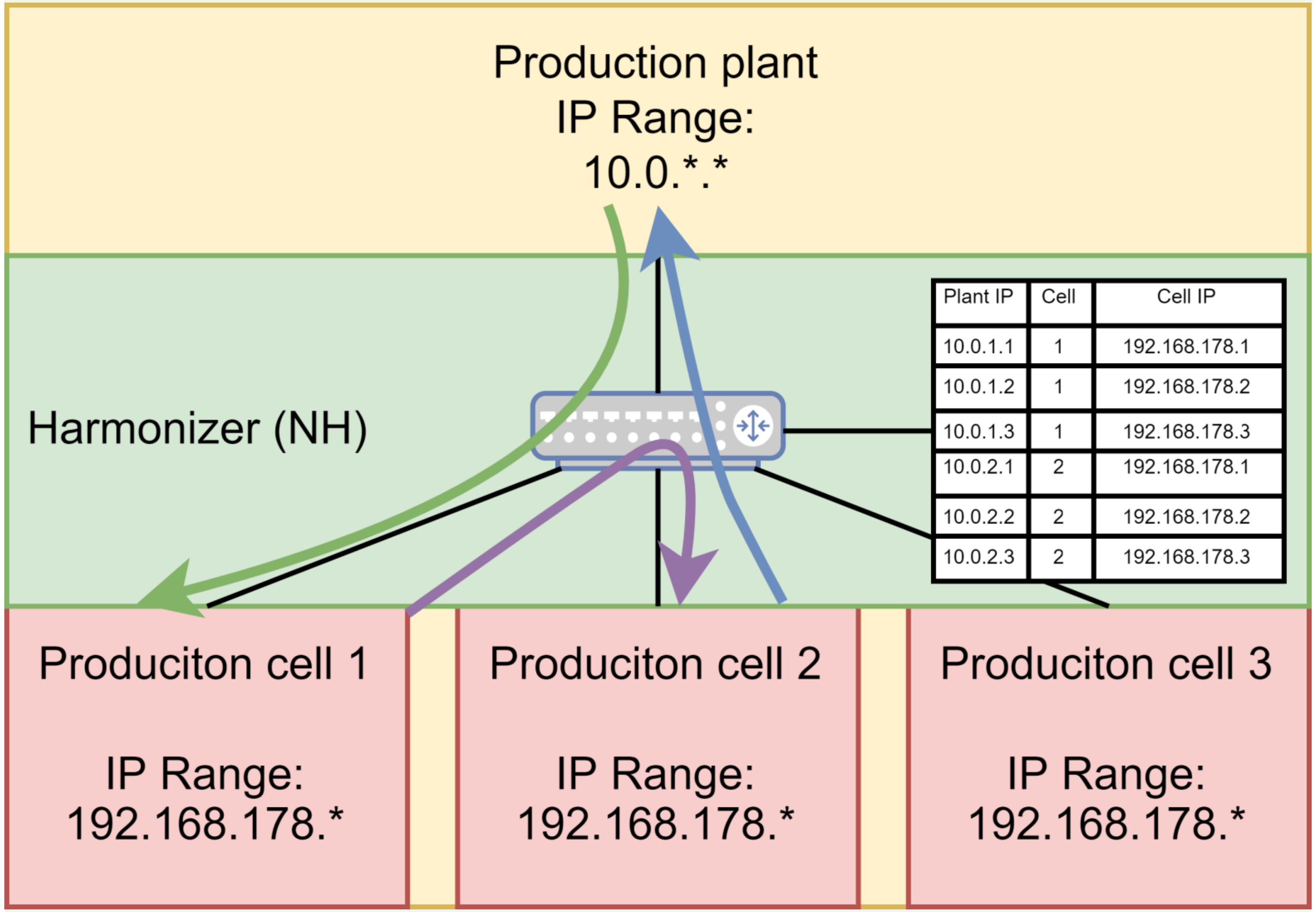
Interworking of Solutions with an example

Deployment Process of TSN application

1. Topology detection & storing (MoNC → Kubernetes)
2. Application request with communication requirements (User → Kubernetes)
3. Application Scheduling and Stream definition (QoS-Scheduler)
4. Stream request (QoS-Scheduler → MoNC)
5. Stream embedding (MoNC)
6. Application deployment (Kubernetes)

SIEMENS

Solution 1: Network Harmonizer



Plant IP	Cell	Cell IP
10.0.1.1	1	192.168.178.1
10.0.1.2	1	192.168.178.2
10.0.1.3	1	192.168.178.3
10.0.2.1	2	192.168.178.1
10.0.2.2	2	192.168.178.2
10.0.2.3	2	192.168.178.3

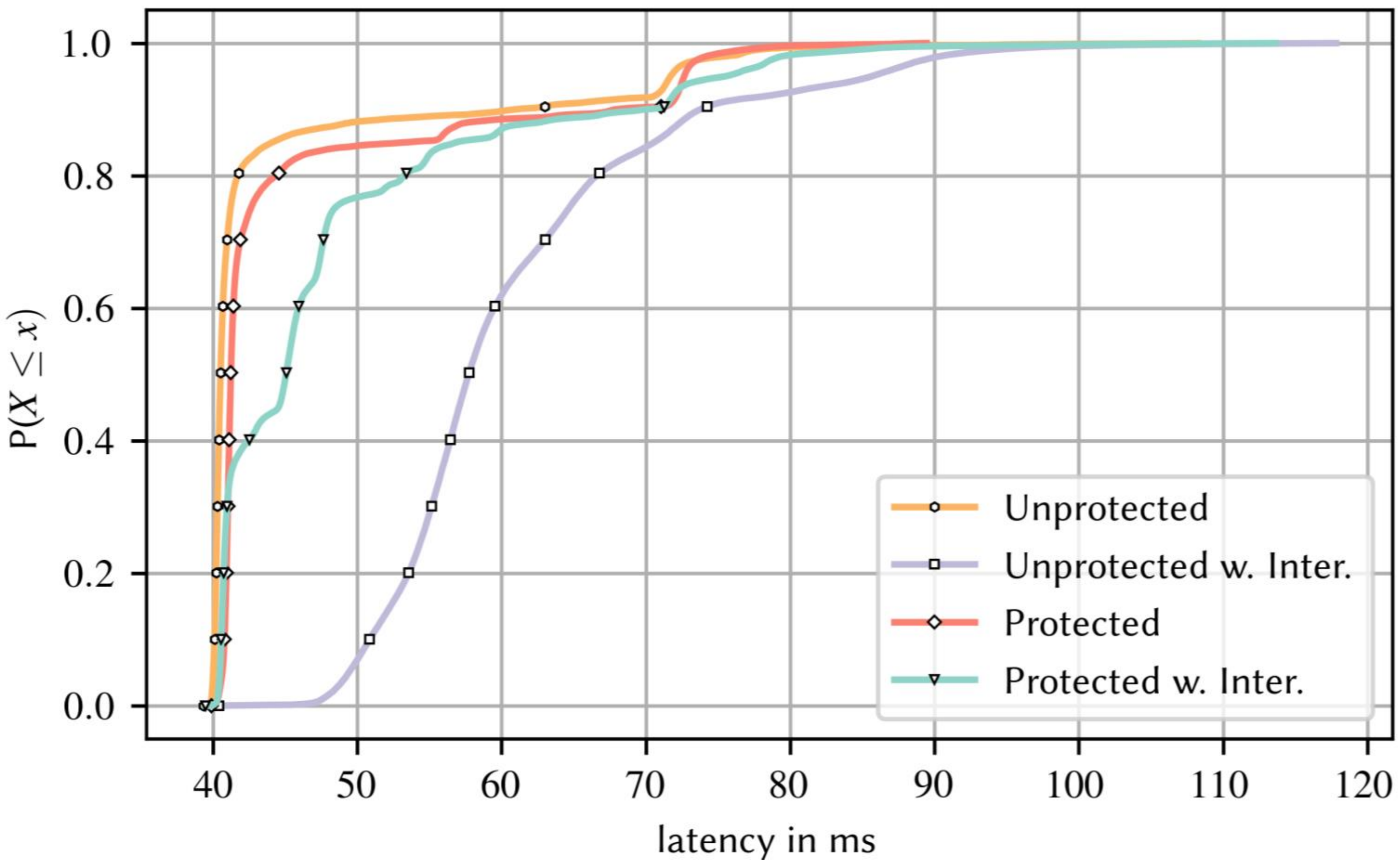
Problem 1: Connect production cells with Data-Link layer access and low configuration overhead.

Function:

- „Fully transparent Router“
- Substitutes IP and MAC Addresses of passing packets.
- Substitutes sender-IP and target-IP of ARP packets
- Connecting production cells with identical IP/MAC configurations.

SIEMENS

Evaluation – Interference Measurements



No Interference:

- Almost same Latency

With Interference:

- Unprotected Stream has Higher latency

→ Streams protected by PSFP

SIEMENS

Conclusion

Proof of concept shows

- Kubernetes adds small latency overhead
- QoS guarantees between Kubernetes Nodes possible

Next Steps:

- Investigate real-time Linux systems to reduce jitter
- Implement other Time-Sensitive Networking Algorithms