tenAsys®

TSN RTOS Integration

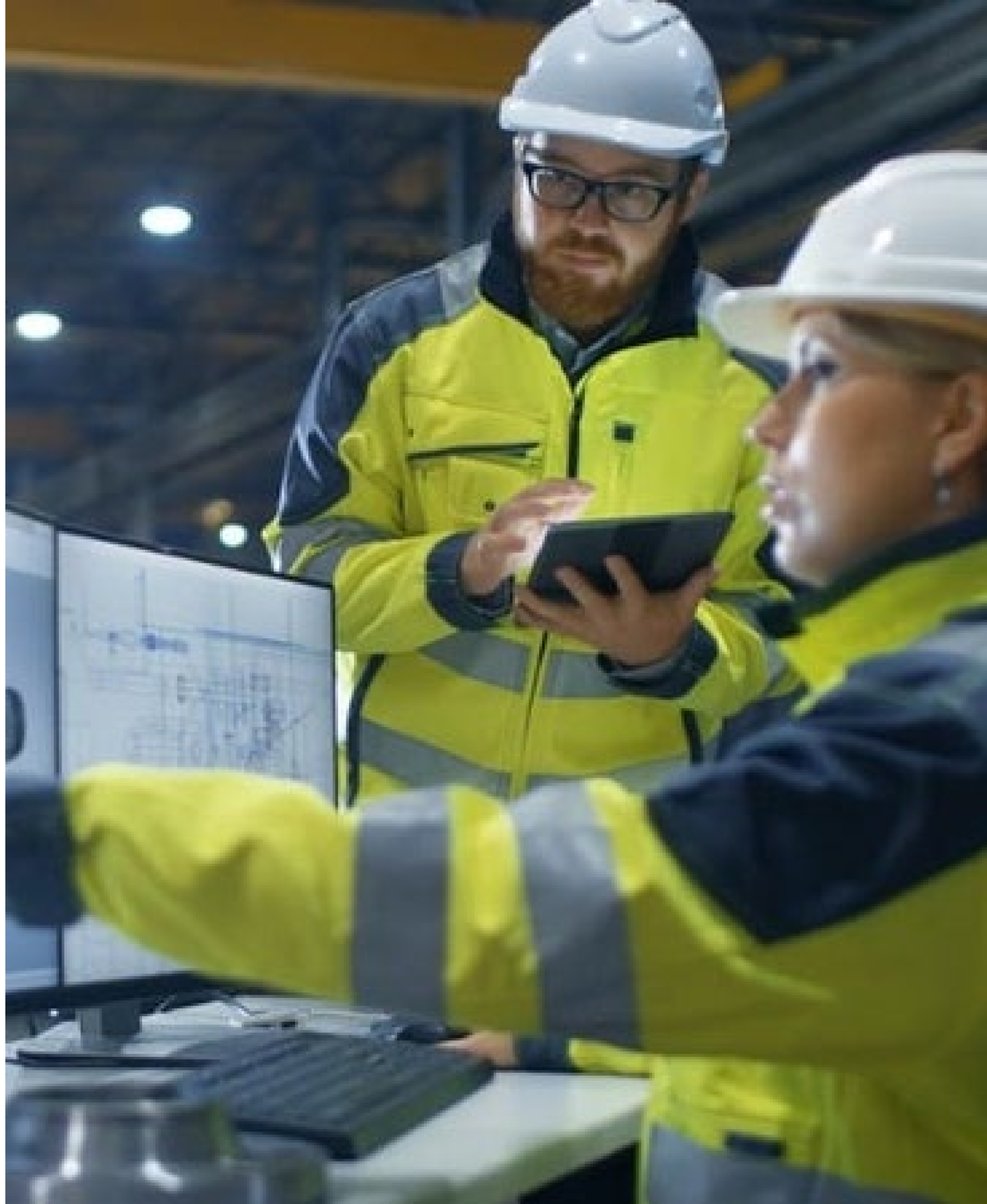
A Resource-Based Approach

Christopher Main & Joel Morrissette

TenAsys Corporation

Florian Frick

ISW, Universität Stuttgart



Agenda

- TSN: Potential and Challenges
- TSN from a User's Perspective
- A Resource-Based Approach
- Application Example
- Perspectives and Outlook

Time-Sensitive Networking

A Key Enabler for Digitalization

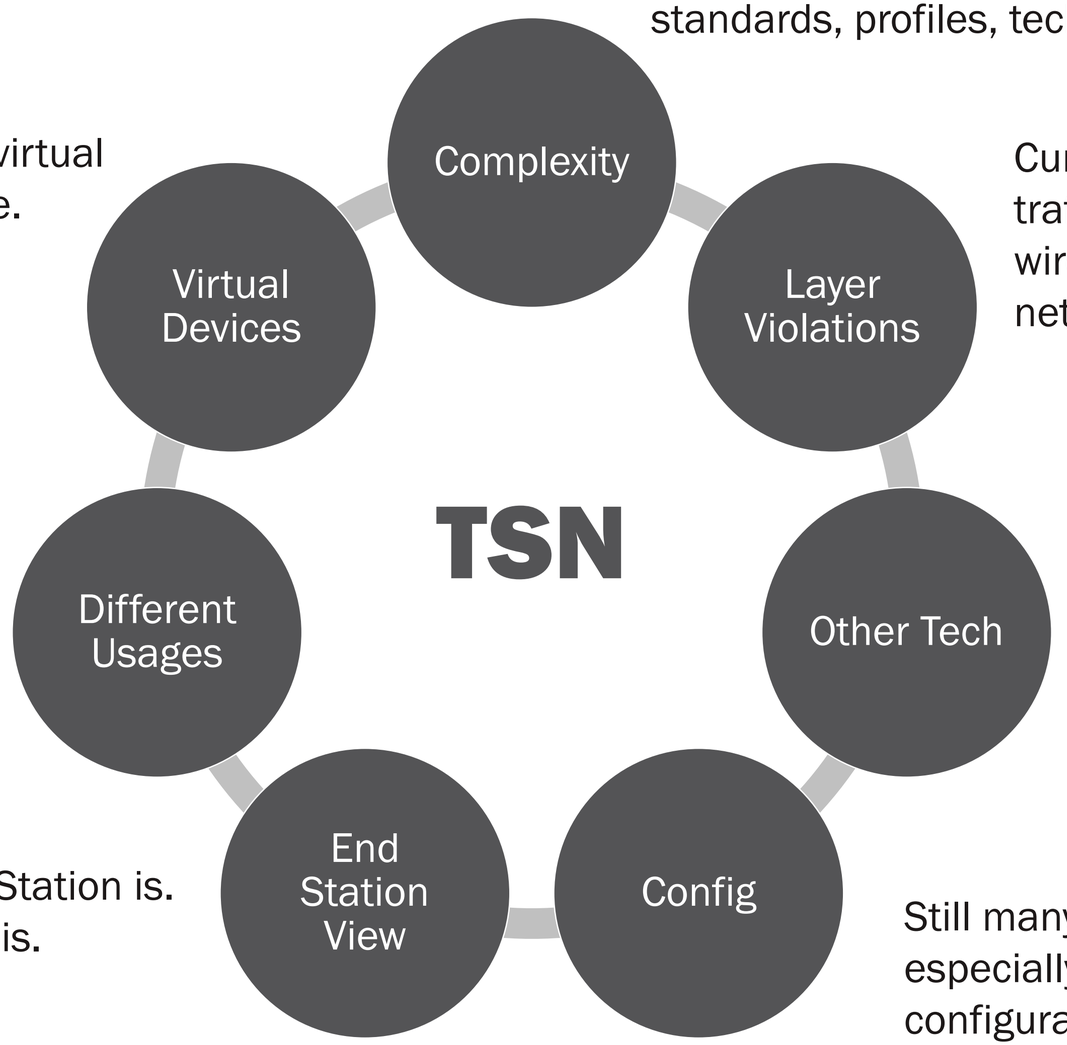
- Digitalization is seen as the key development for the future of industrial automation
- Many companies have declared Industry 4.0 as a strategic goal
- Today’s networks are insufficient, converged real-time networks required
- TSN is an accepted solution across different industries

TSN Implementation Challenges

Complexity arises from the proliferation of standards, profiles, technologies, and hardware.

Virtualization results in multiple virtual End Stations on a physical device.

Currently real-time solutions network traffic is not in-layer, tend to own the wire, and do not often co-exist with other network applications.



There are many ways to use TSN including usage for different fieldbus over TSN solutions. Not all are necessarily cooperative with other use-cases.

Other technologies with similar features include

- 5G/6G
- Wireless TSN
- DetNet

No unified answer as to what an End Station is. TSN standards have not addressed this.

Still many open questions, especially about End Station configuration

Agenda

- ✓ TSN: Potential and Challenges
- ✓ **TSN from a User's Perspective**
- ✓ A Resource-Based Approach
- ✓ Application Example
- ✓ Perspectives and Outlook

User Perspective: Need

Who is the user?

- Application engineer, not network engineer
- Should everybody be a TSN expert?
  - Ideally nobody is – users should be aware of capabilities, but are content experts not network experts.
  - Do we even care if TSN is underlying our converged network? Or something else? If it delivers what we need in terms of connectivity?

User Perspective: Workflow

What is the user really interested in?

- Time-sensitive communication, not networking implementations, h/w features, etc.
  - As defined by application requirements
- Modifying existing system to interact with TSN-connected devices
  - E.g., how do I migrate a controller from EtherCAT-connected devices to TSN?
- Virtualization of connection – move from dedicated connection to connected via TSN
  - Connect controllers to devices across converged network

Agenda

- ✓ TSN: Potential and Challenges
- ✓ TSN from a User's Perspective
- ✓ **A Resource-Based Approach**
- ✓ Application Example
- ✓ Perspectives and Outlook

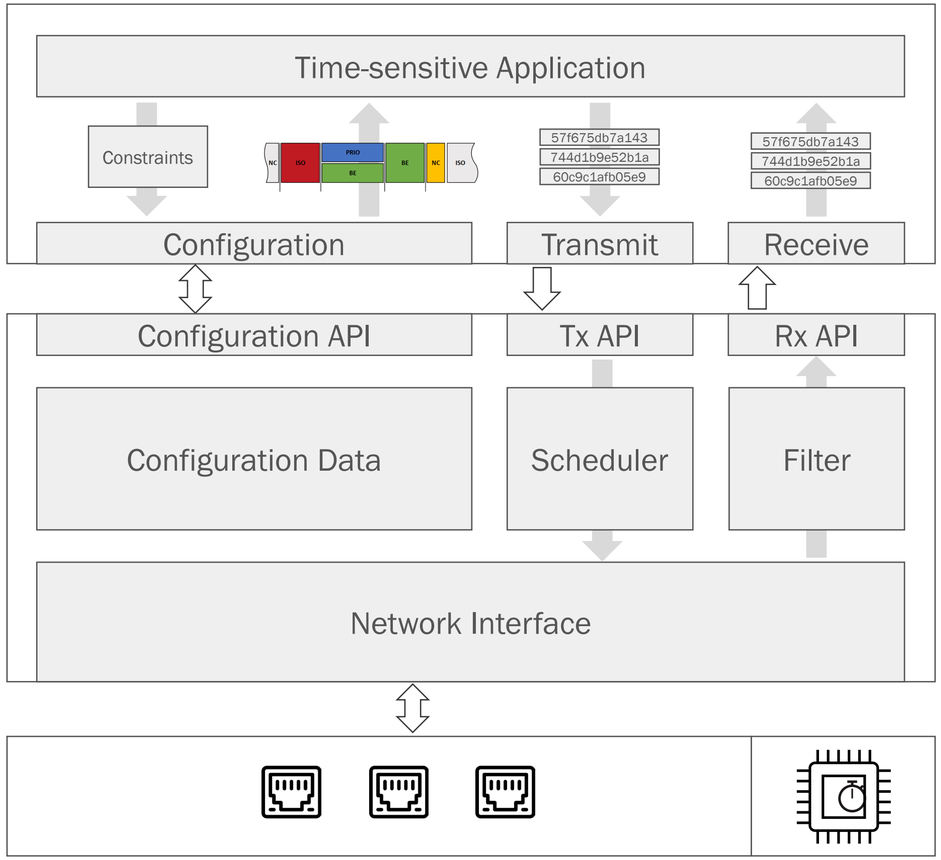
Resource-based Approach

Abstraction of the TSN Network

- A configured TSN network provides communication resources
  - Streams
  - Traffic classes (e.g. Best Effort, Shaped)
  - => Derived from the system schedule and all application requirements
- But what does the user (or application) see?
  - List of resources and capabilities
  - Actual resources matched to application requirements
- Need to understand resources, not standards
- Application is transport-agnostic, but QoS-specific.

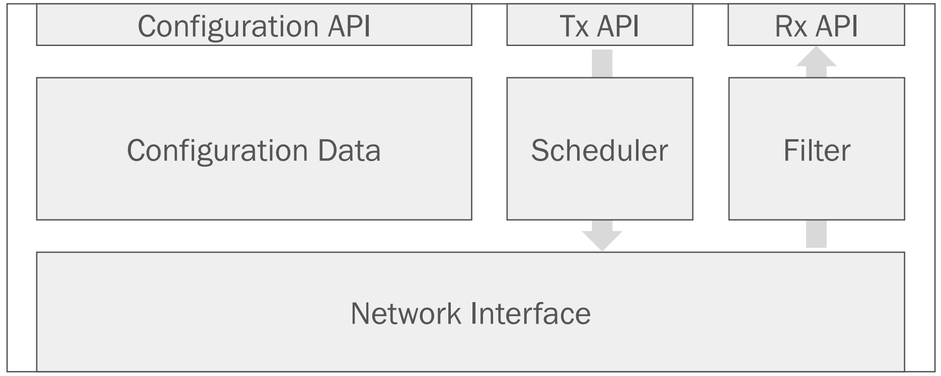
Three Layers

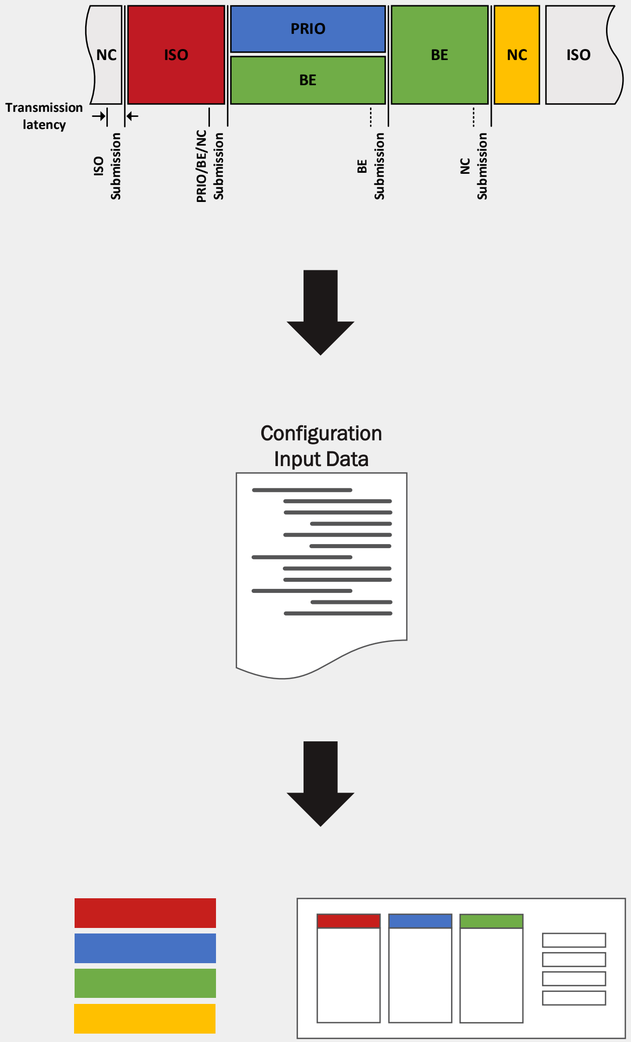
- Application
  - There is no “TSN application”, but applications have certain deterministic requirements for their communication which TSN may be able to resolve
- Technology abstraction and aggregation
  - TSN, and/or other technologies
- Hardware
  - The hardware solutions are not complete TSN solutions.
  - Network interfaces have different capabilities



Abstraction Logic

- The Configuration Data is used by the TX scheduler to determine when packets are to be transmitted via the network interface
- Also used by the Rx interface to know where to deliver received packets from the network interface
- How the configuration data is presented is key to the implementation.

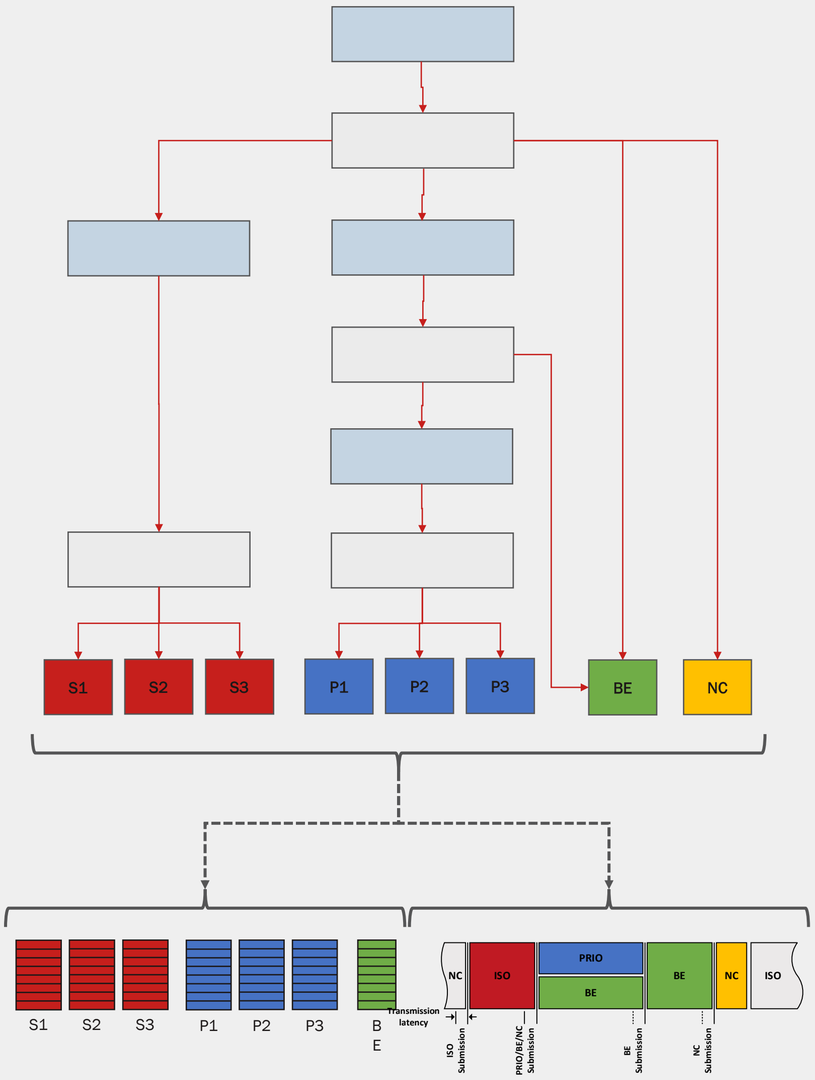




Resource Creation

Mapping system capabilities to application requirements

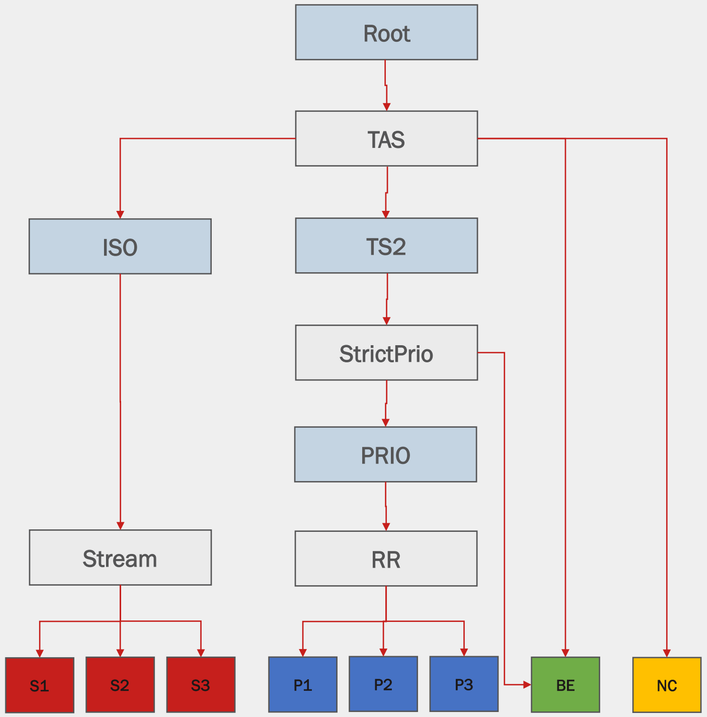
- There is a network schedule based on the chosen configuration approach
- Configuration data is generated to build a list of resources available to end stations
- Each resource is abstracted to a local resource unit
- The local configuration is derived from the list of abstract resource units
- The resource model is independent of the source of the configuration data



Resource Map

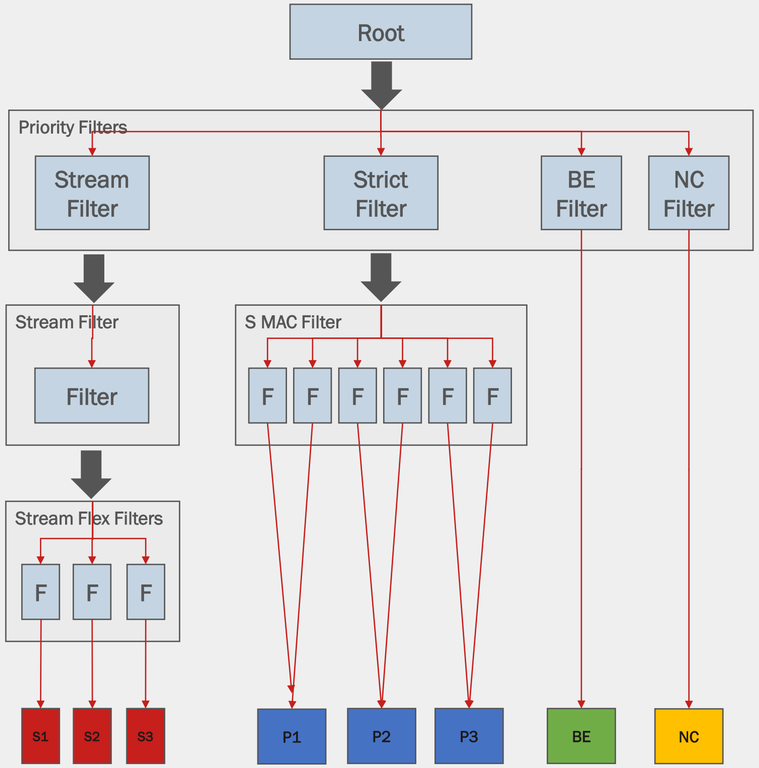
Mapping system capabilities to application requirements

- Configuration data is used to build a map of resources available to user/applications.
- The map gives a unified view of:
  - Hardware capabilities
  - Standards
  - Additional internal traffic concepts
    - e.g. time-aware with more than eight traffic classes
- Leaf nodes are now sufficient to describe both transmit and receive characteristics



Transmit CRU Map

- Tx Communications Resource Units (CRUs)
  - Represents an interface, a traffic class, or a stream
  - The Root represents all traffic transmitted through an interface
  - Each CRU may be subdivided by applying a shaper
  - Each leaf is visible to the application
- Shapers
  - Generate new CRUs
  - Shaper types include
    - Time-Aware Shaper
    - Stream Shaper
    - Priority-based shaper



Receive CRU Map

- Rx Communications Resource Units (CRUs),
  - Represent an interface, a traffic class, or a stream
  - The Root represents all traffic received through an interface
  - Sorted by a filter
  - Each leaf is visible to the application
- Filters
  - Subdivide traffic according to filter rules
  - Different filters are applied: Priority, Streams, MAC address, etc.
  - Defines a hierarchy that results in correct data delivery

Datapath Details

- Application Data Interface
  - Instantiates a communications endpoint from a CRU
  - Send: supply as required: destination MAC address, frame type and payload.
  - Receive: inbound frames are filtered and delivered to the appropriate endpoint
  - Streams: static buffer(s) supplied by system; user/application must fill buffer before launch cutoff time
- HW-SW
  - Tx schedule is built in software, uses interface-dependent hardware features to offload scheduling as available
  - Hardware filtering of inbound traffic also used to the extent it is available
  - H/W offload not directly visible to user but may affect overall performance

Configuration: Basic

- Different entry points for defining configuration
  - Manual (local tool)
  - Predefined profiles, e.g. loaded from text file.
  - Centralized configurations applied by networked configuration component
- A configuration API is available to all these methods
- In each case an application will obtain configuration data then call the API to generate the CRUs.
  - An agent may read a static configuration from a text file and apply it via the API to create the CRUs.
  - An agent may make a request to a CUC then apply the response via the API to create the CRUs.
- Such agents may be created on a per-application basis for now until standards are firmer.

Configuration: Mapping

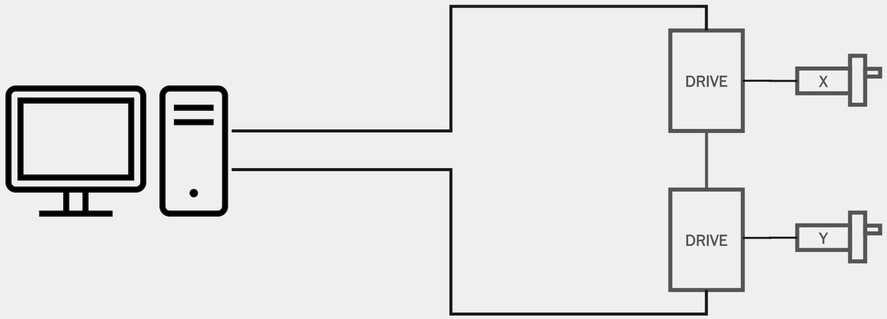
- How is the internal configuration derived?
  - The API will create CRUs from the supplied parameters (QoS, offset, priority, as required) and shaper objects as required.
- The user/application API enumerates the leaf CRUs and instantiates the requested resource.
- Beyond simplicity this gives the user
  - Portability
  - Flexibility
  - Extensibility

Agenda

- ✓ TSN: Potential and Challenges
- ✓ TSN from a User's Perspective
- ✓ A Resource-Based Approach
- ✓ **Application Example**
- ✓ Perspectives and Outlook

Basic Application Example

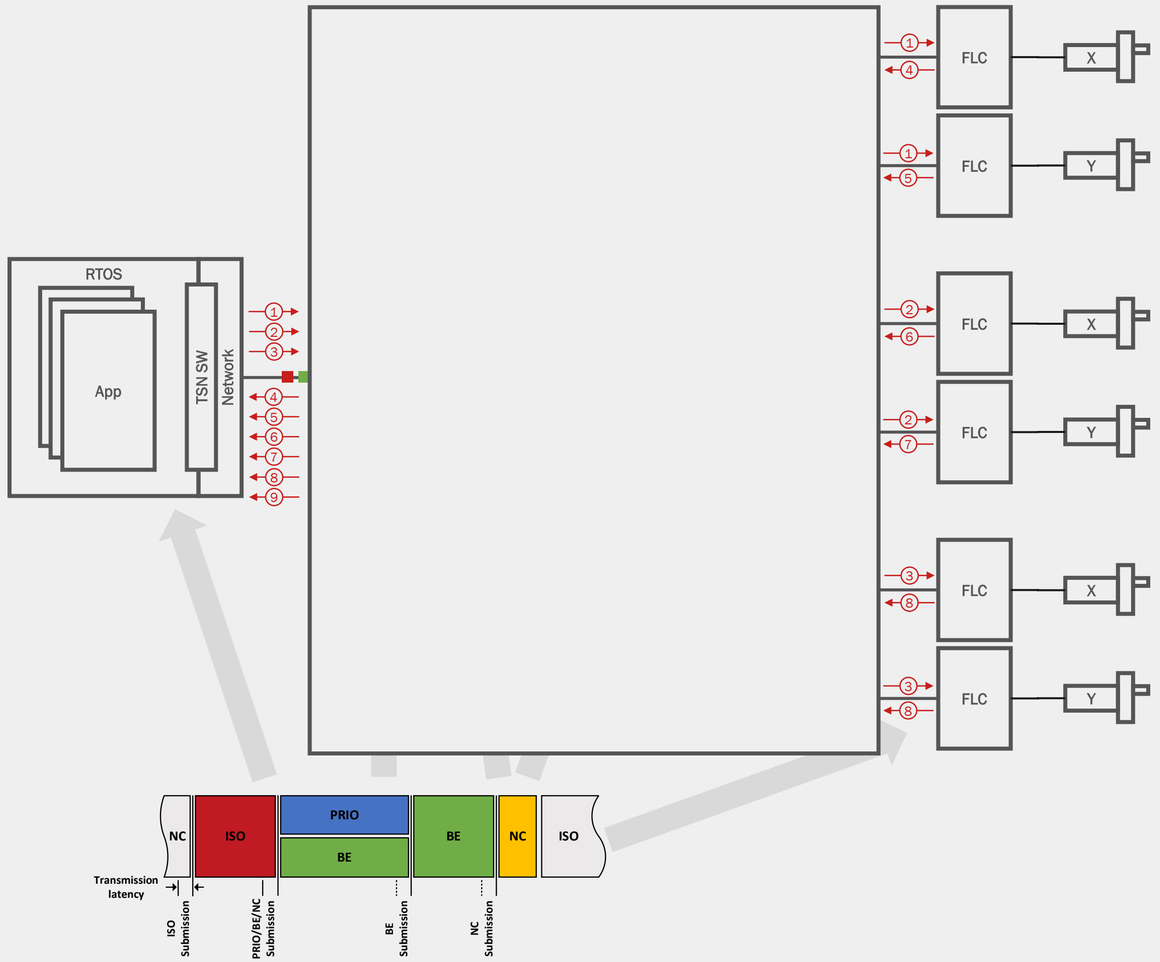
- A controller runs a single application responsible for controlling two drives
- The controller is responsible for all schedule information and is solely responsible for ensuring QoS constraints



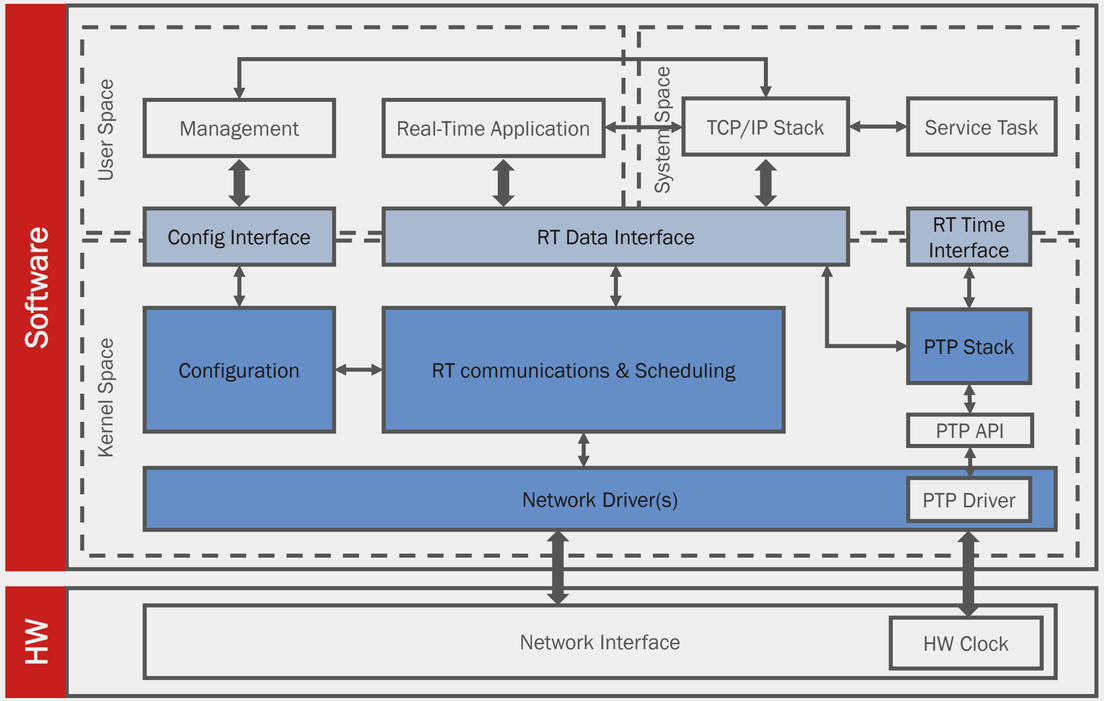
TSN Application Example

Scaling to multiple virtual controllers.

- Multiple virtual controllers on same host
- Each controller runs a single application responsible for controlling two drives
- System components (e.g. CUC/CNC) responsible for creation of schedule according to QoS requirements of applications.



System Architecture (Simplified)



Agenda

- ✓ TSN: Potential and Challenges
- ✓ TSN from a User's Perspective
- ✓ A Resource-Based Approach
- ✓ Application Example
- ✓ **Perspectives and Outlook**

Eased Adoption

Abstraction

By abstracting most of the details of the network, the protocols, hardware, etc., the user is left with only the resources required to interact with devices in the converged network.

Simplification

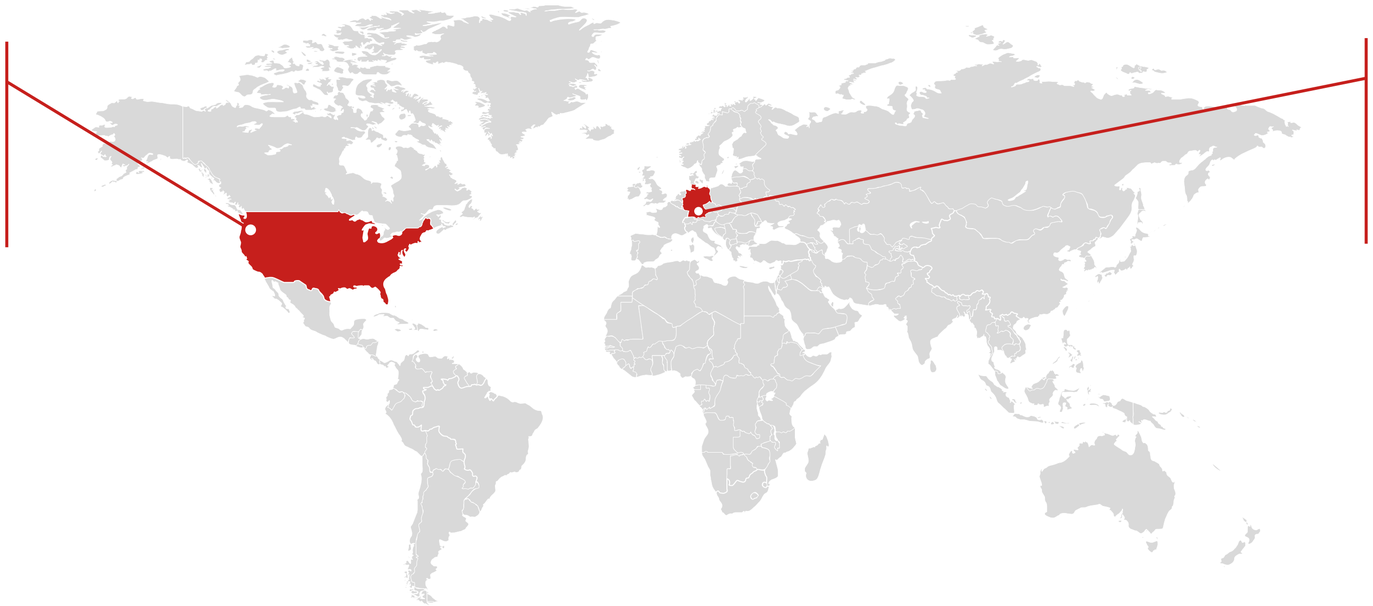
Simplification is combined with adaptability and extensibility to easily adapt to future changes, features and technologies

Questions

Christopher Main

President & CTO

chris.main@tenasys.com | www.tenasys.com



TenAsys Corporation
1400 NE Compton Drive, Suite 301
Hillsboro, OR 97006
USA
+1 503 748 4720

TenAsys Europe GmbH
Hans-Stiessberger-Str. 2b
85540 Haar / Muenchen
Deutschland
+49 (89) 46 1498 45