



# Abstracting Time-Sensitive Networking Complexity to Accelerate IT/OT Convergence

**Christopher Main**

CTO, TenAsys Corporation





# Agenda

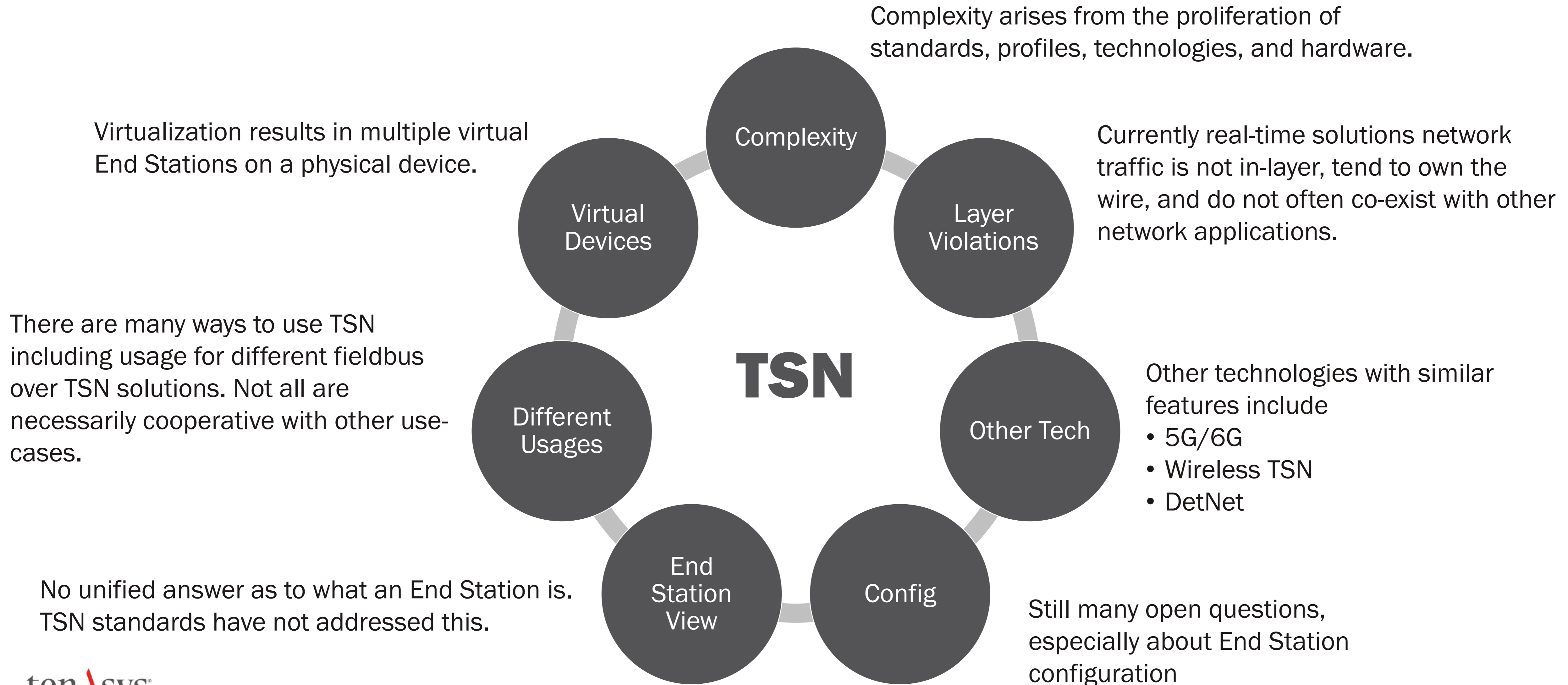
- ✓ **TSN: Potential and Challenges**
- ✓ TSN: Converged Networking Hero Use-Case
- ✓ A Resource-Based Approach
- ✓ Application Example
- ✓ Perspectives and Outlook

# 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



# Agenda

- ✓ TSN: Potential and Challenges
- ✓ **TSN: Converged Networking Hero Use-Case**
- ✓ A Resource-Based Approach
- ✓ Application Example
- ✓ Perspectives and Outlook



# Enabling Industry 4.0

## A Hero Use Case

Converged networking is a fundamental enabler for Industry 4.0

and is required by Software Defined Manufacturing

which is enabled by TSN running on commodity hardware providing implementation flexibility, simplicity, and lower TCO.





# Agenda

- ✓ TSN: Potential and Challenges
- ✓ TSN: Converged Networking Hero Use-Case
- ✓ **A Resource-Based Approach**
- ✓ Application Example
- ✓ Perspectives and Outlook

# Resource-based Approach

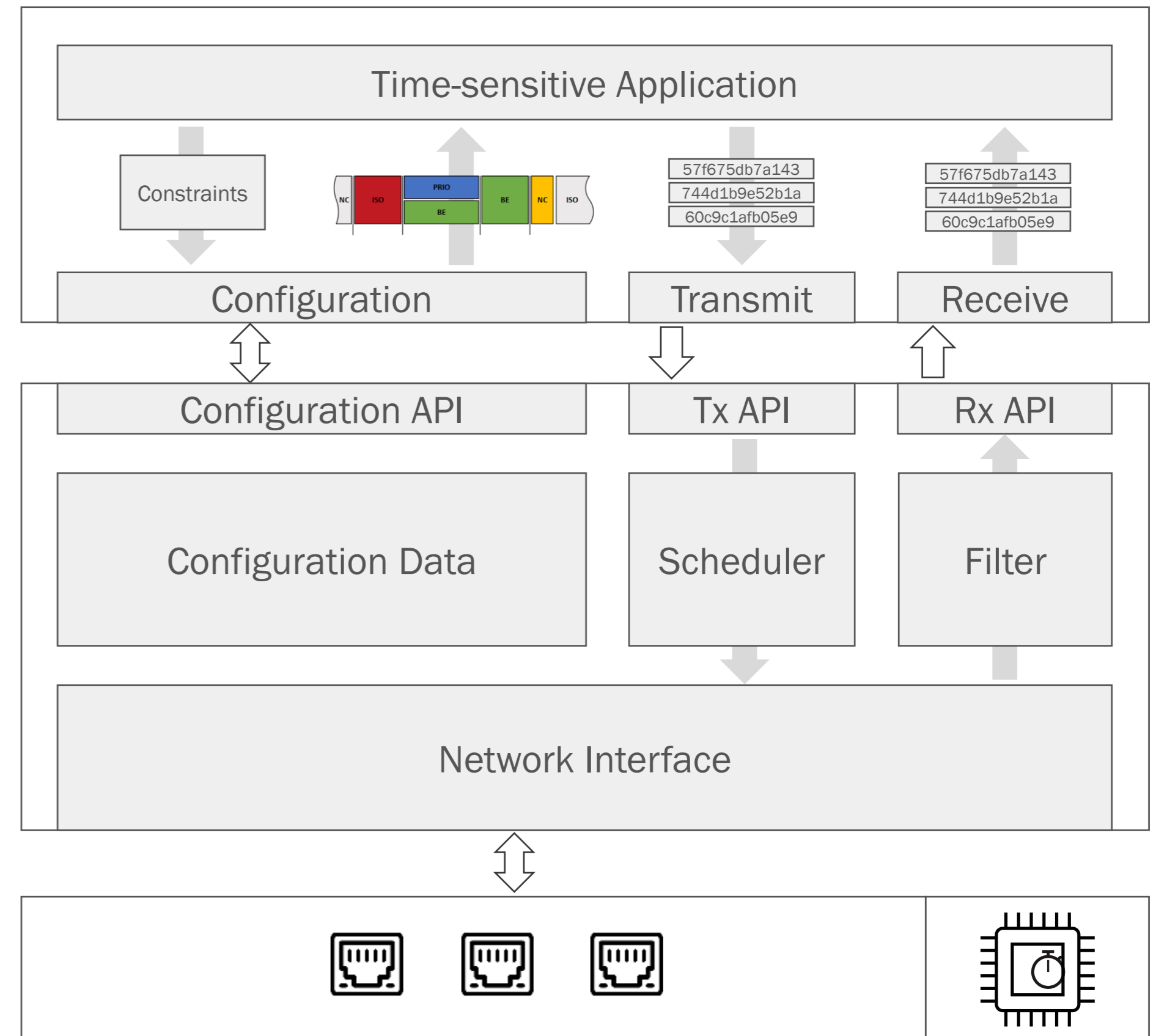
As viewed by a controller on a TSN network

- A configured TSN network provides communication resources derived from the system schedule and all application requirements:
  - Streams
  - Traffic classes (e.g. Best Effort, Shaped)
- 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.



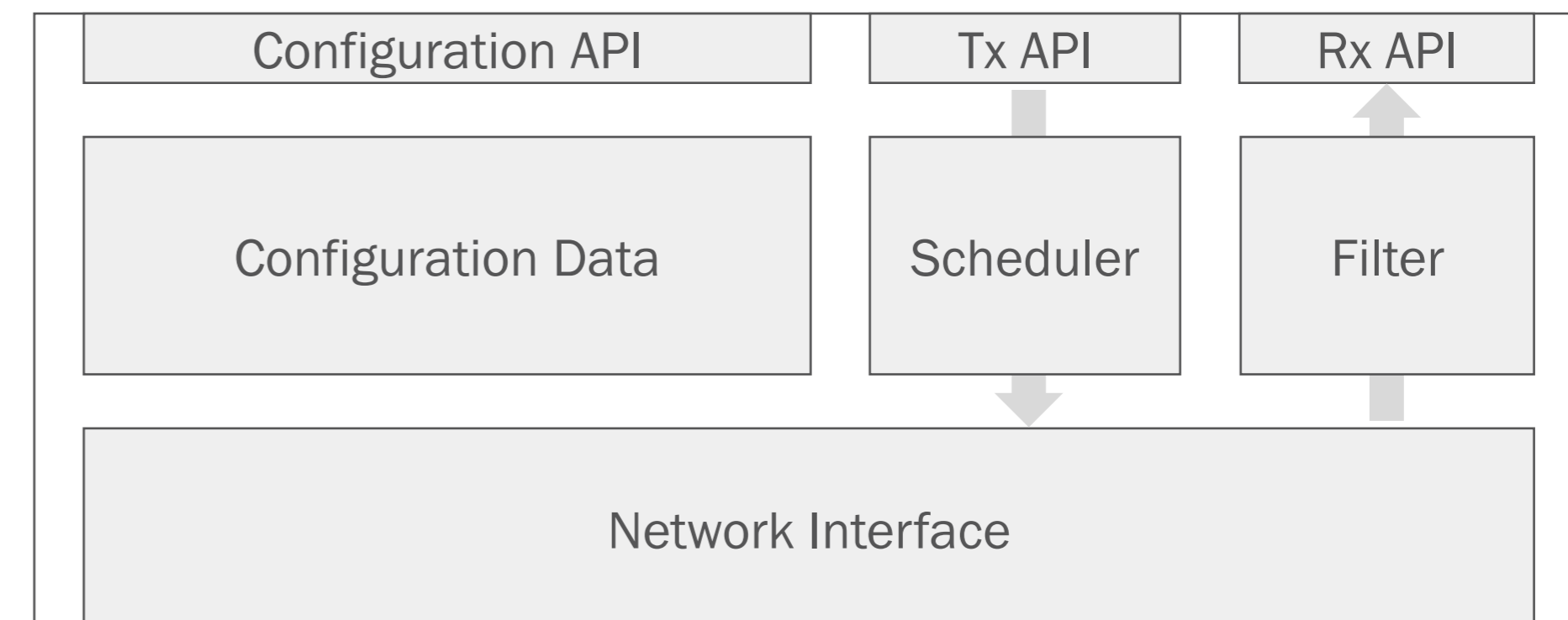
# Three Layers

- 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

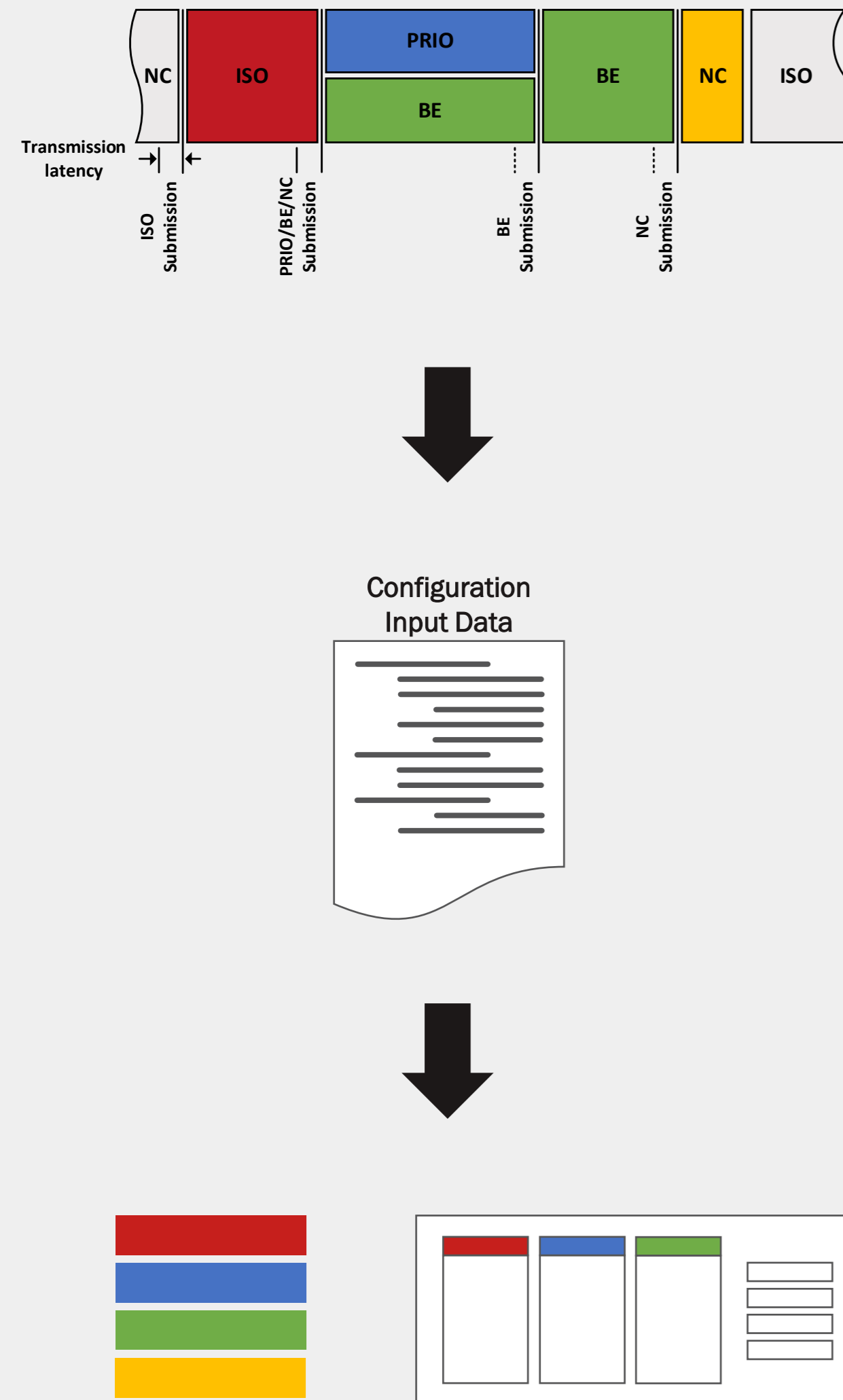


# Abstraction Logic

- 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

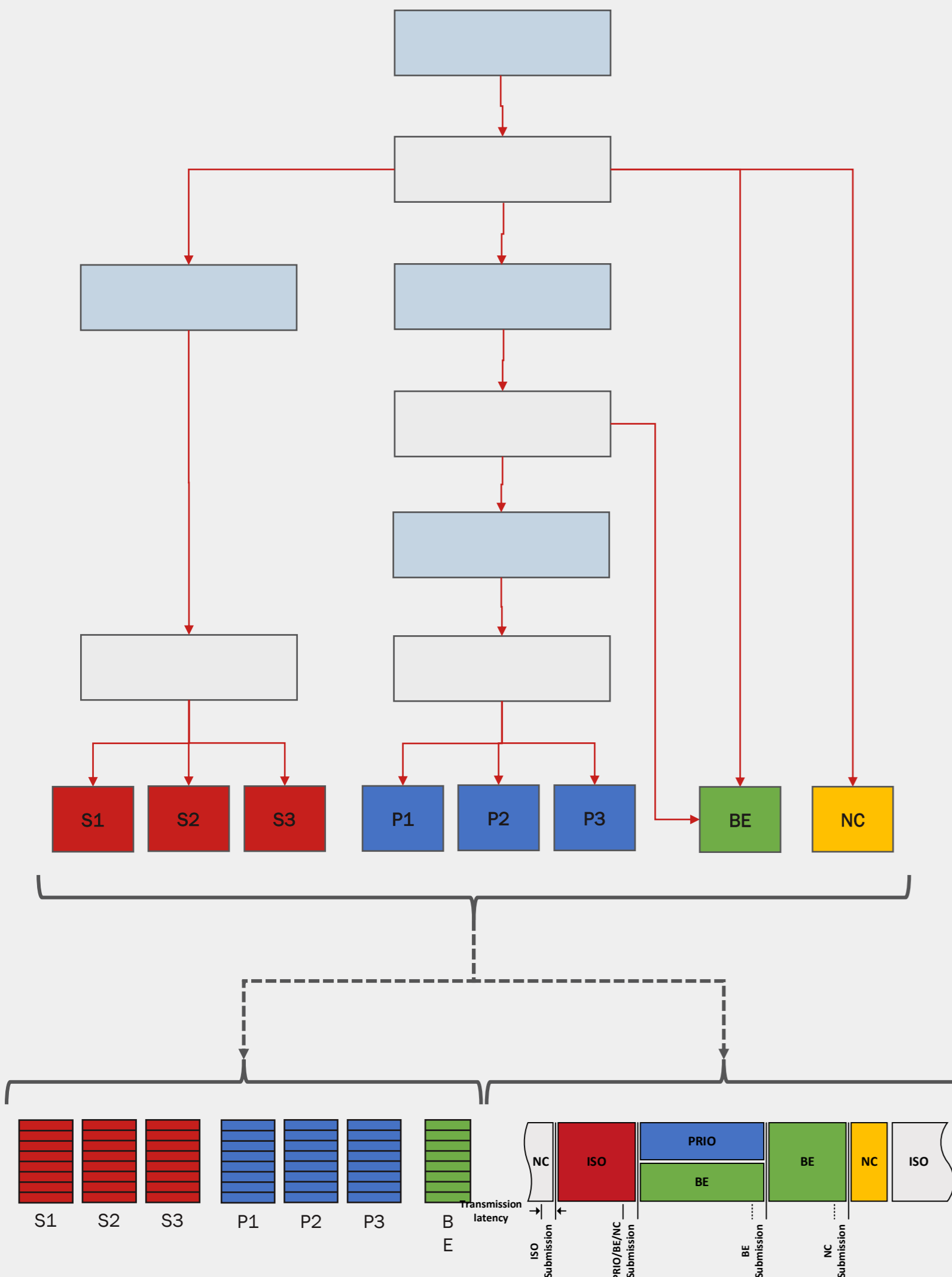
Mapping system capabilities to application requirements

- 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

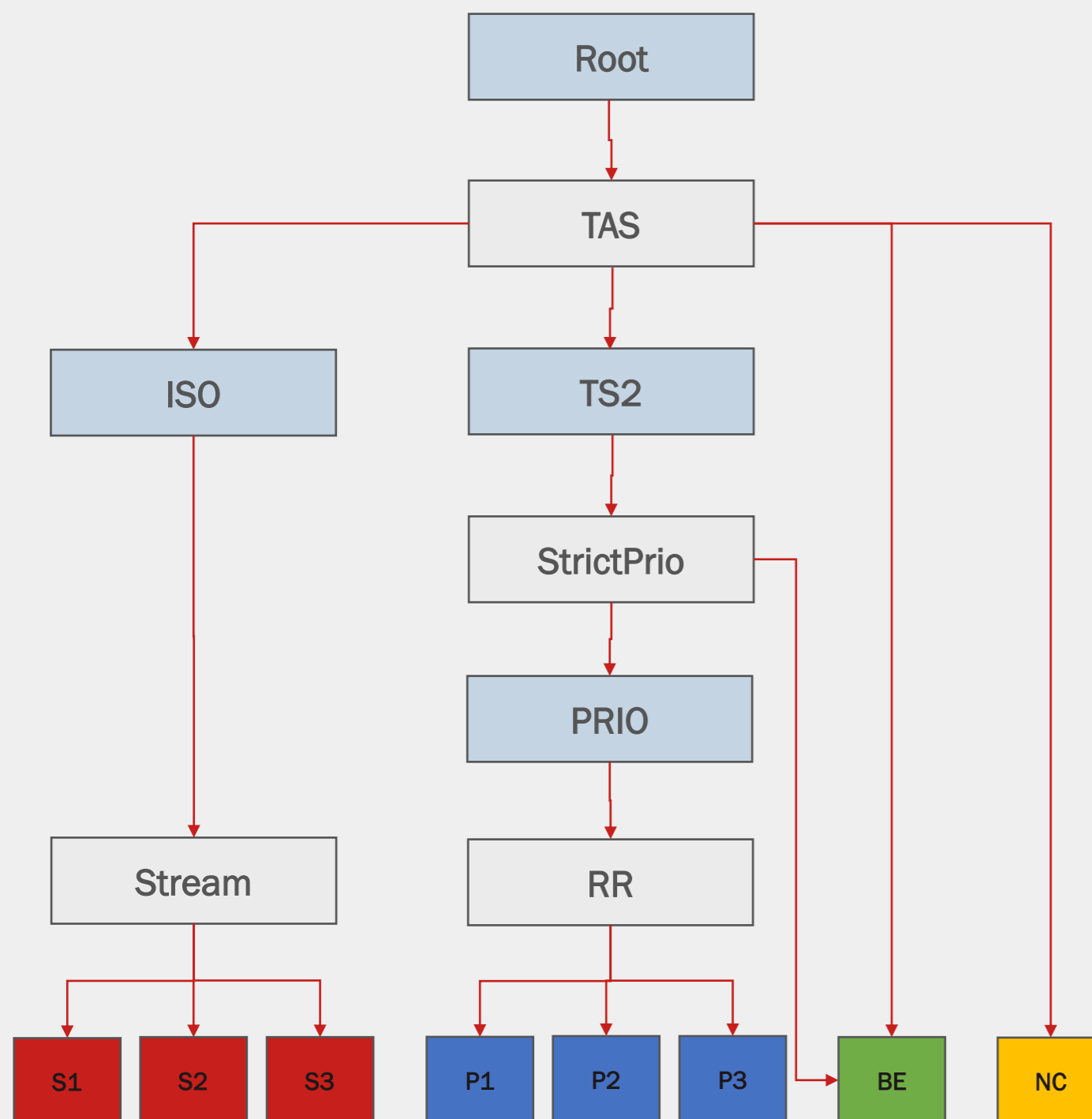
Mapping system capabilities to application requirements

- 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



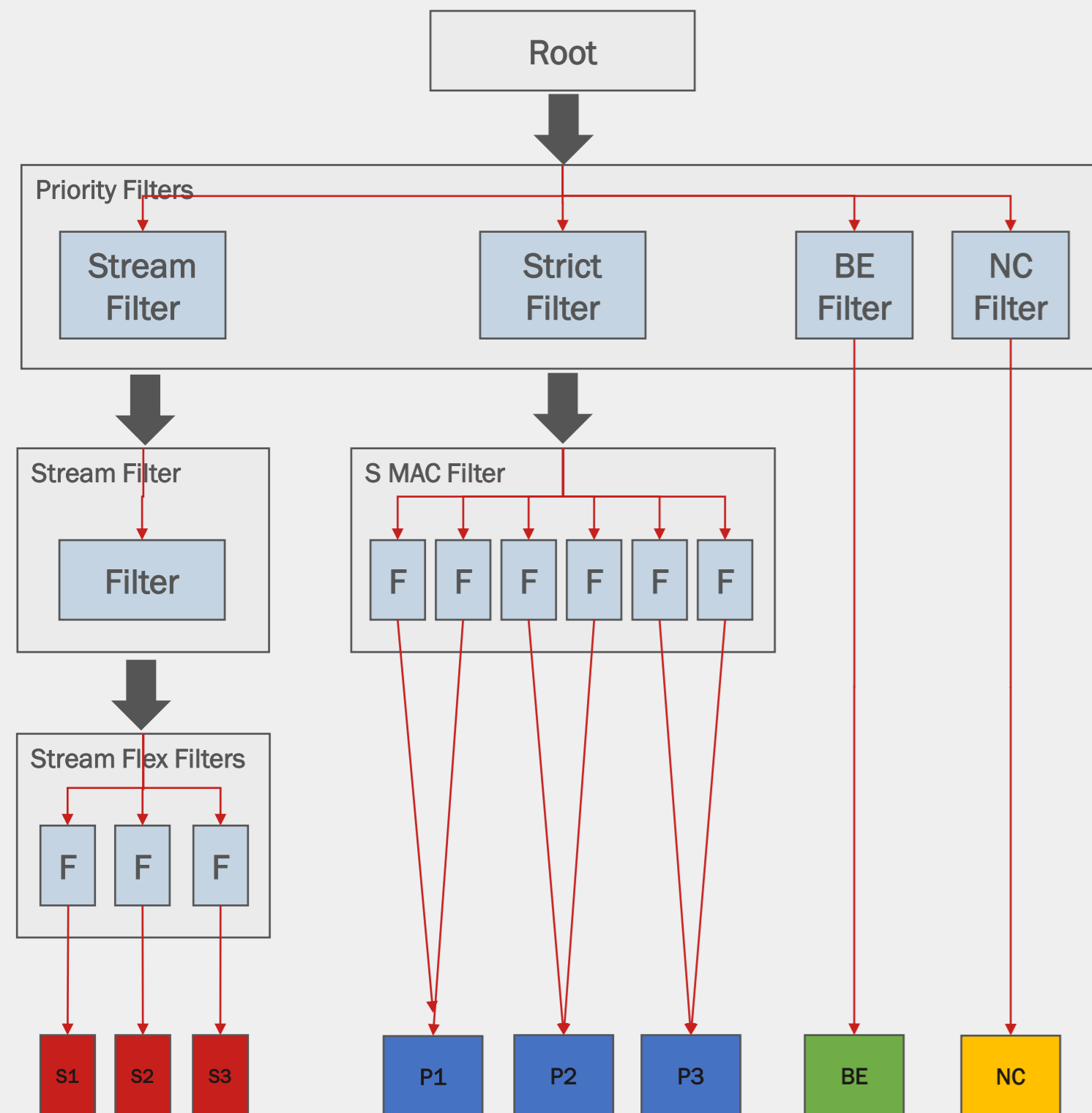


# Transmit CRU Map



- 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 interface
- 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
- H/W-S/W
  - 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

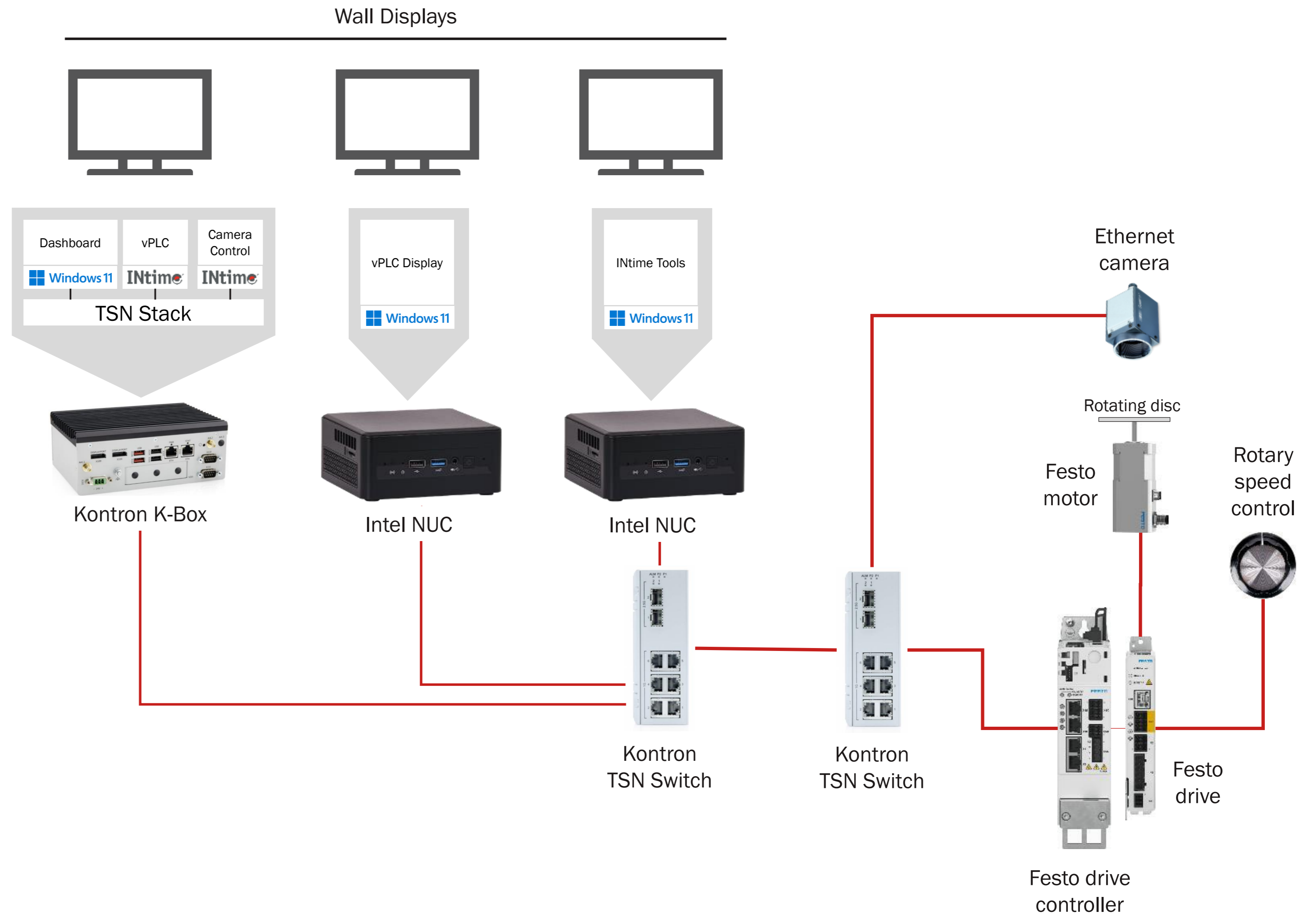
- ✓ TSN: Potential and Challenges
- ✓ TSN: Converged Networking Hero Use-Case
- ✓ A Resource-Based Approach
- ✓ **Application Example**
- ✓ Perspectives and Outlook

# Application Example

Booth Demo: Converged Networking

- Multiple applications using converged TSN network
- Built from off-the-shelf hardware, enabled with the INtime® TSN stack
- Both real-time control and user interfaces hosted on a single system, partitioned to host multiple OSes

Hall 4, Stand 4-458



# Agenda

- ✓ TSN: Potential and Challenges
- ✓ TSN: Converged Networking Hero Use-Case
- ✓ A Resource-Based Approach
- ✓ Application Example
- ✓ **Perspectives and Outlook**



# Eased Adoption

In order to enable Converged Networking to implement our Hero Use Case, we need...

## 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.

## Innovation

Through a unified platform that integrates a full TSN implementation with the INtime RTOS, developers can more rapidly iterate and deploy TSN solutions ready for Industry 4.0 and SDM.

# Christopher Main

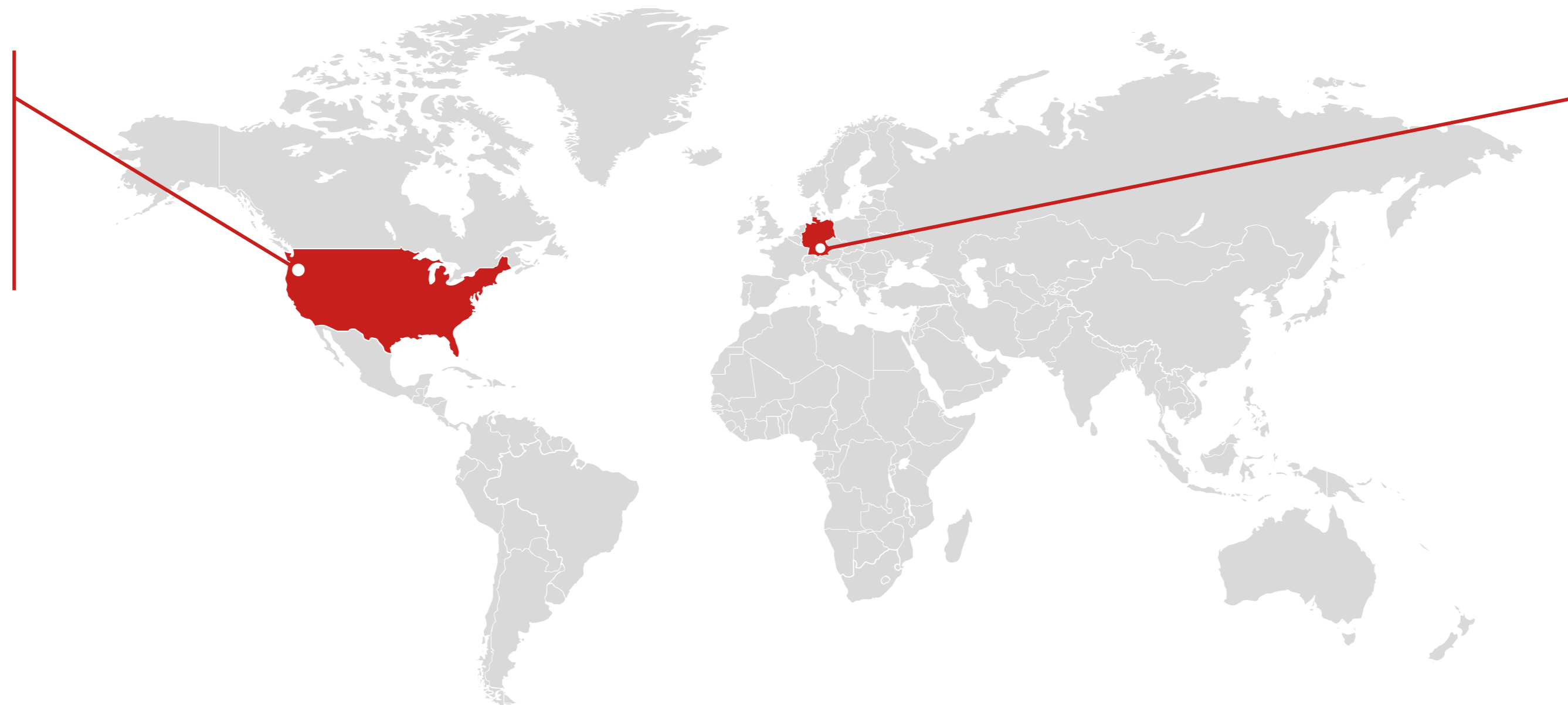
President & CTO

[chris.main@tenasys.com](mailto:chris.main@tenasys.com) | [www.tenasys.com](http://www.tenasys.com)

Visit TenAsys in **Hall 4, Stand 4-458**

## 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