KRYO

Real World Requirements

Context: Hot War in Europe

- The war of one-day ideas
  - Physical sub-systems change slowly
  - *Everything* else is changing rapidly
  - Observers realizing they are built out for WWII, not for today
- All observers rethinking defensive posture
  - Old: Out-spend on hardware to create advantage
  - New: Software Defined Defense

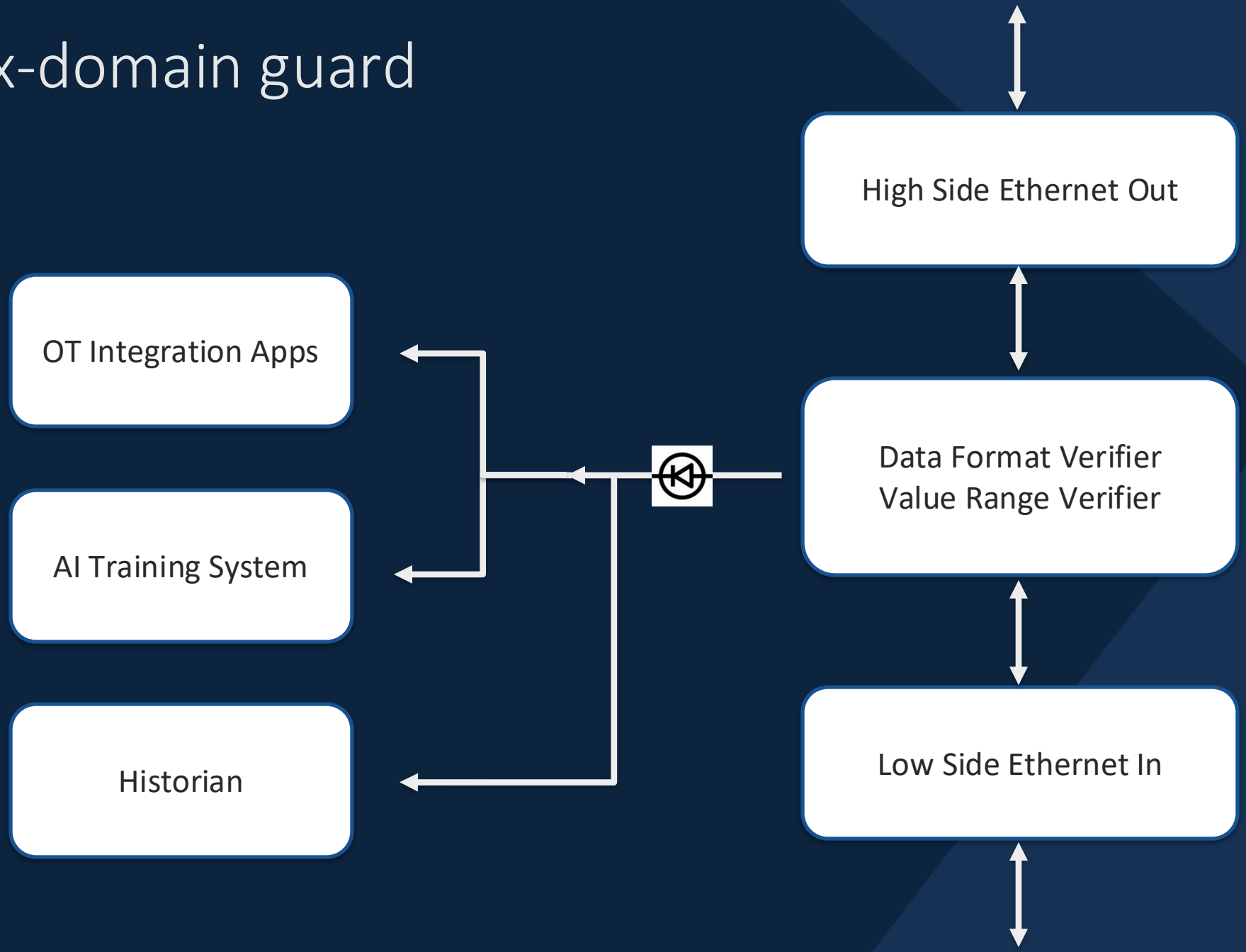
Context: Cold War 2.0

- The war of single-use weapons
  - Offense is so far ahead of defense that there is no defense
  - Mutually Assured Destruction in 10 minutes
  - Cyber weapons once used lose offensive value
- Defensive posture needs a complete overhaul
  - Active assured response is desperately needed
  - OT must embrace no-downtime, assured management
  - Must be simple as it goes down to the municipality level

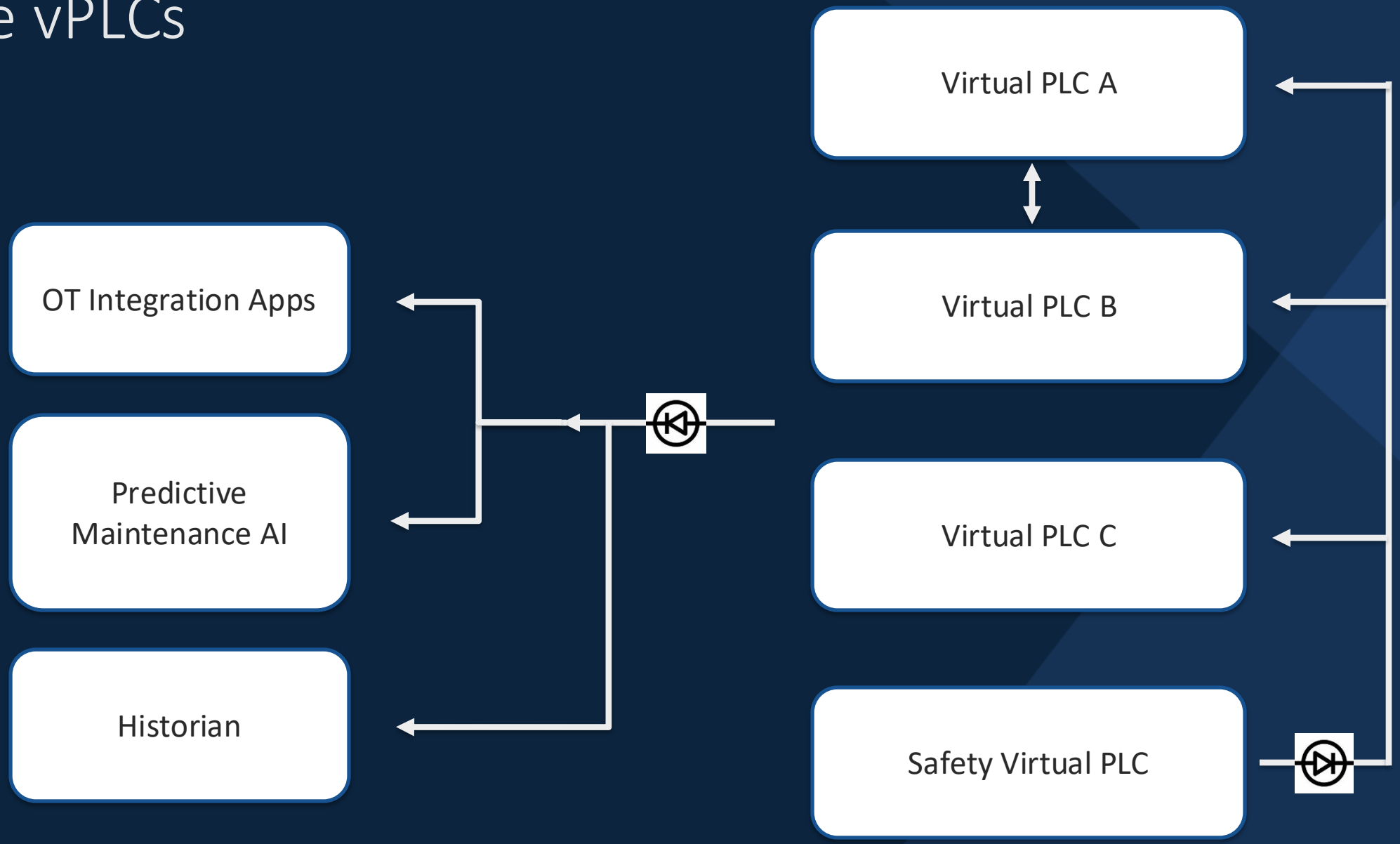
Context: The New Dual-Use

- Old: Defense and Consumer
  - Technologies that have specific defense needs
  - Combined with use in scaled consumer products
  - Both fit into existing philosophies
- New: Modern Hot and Cold War 2.0
  - Use in hot defense
  - Use in cold infrastructural systems
  - Requires massive overhaul down to the municipality lecel

Simple x-domain guard

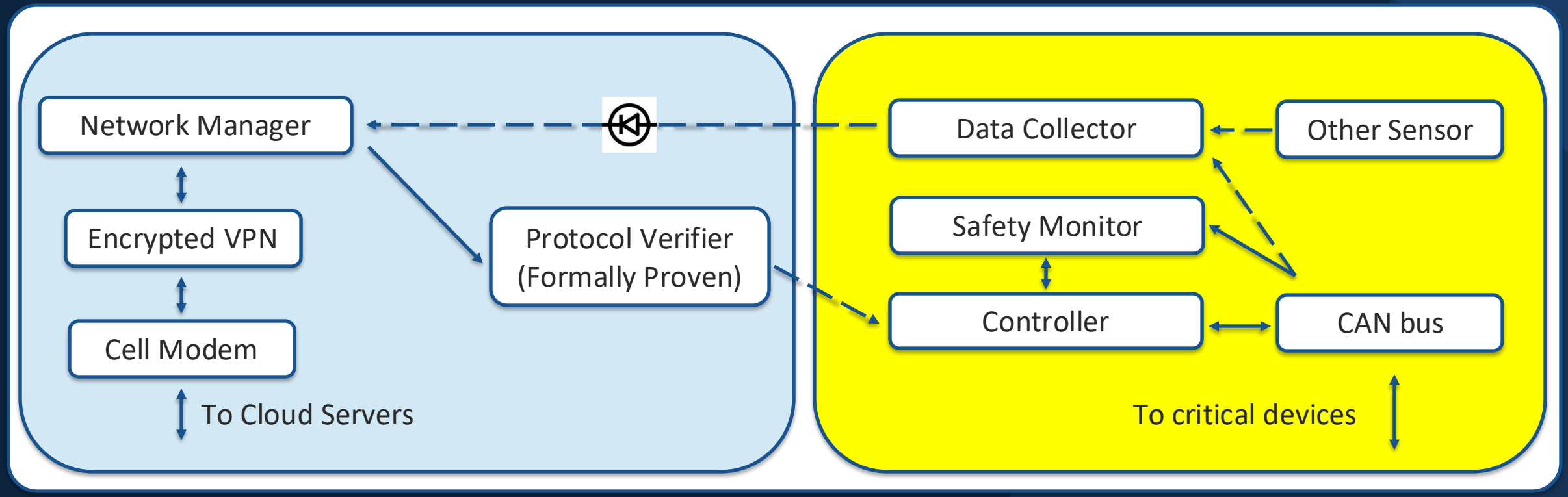


Multiple vPLCs



Think in terms of zones

- High safety vs Low Tolerance
- Rapidly changing software vs. slow and stable
- Regulatory regions



Overall system requirements

- Machines are always on
  - Updates, reactions, allowed zero downtime
  - Missions, designs, even architectures not known until deployed
- Must comply with regulatory regimes
  - ISA 62443, ISO 26262, DO-178x, and many more
- Change is assured
  - The only thing we know is that we don’t know what a machine will encounter when it is made

Kernel Requirements

- Verified Isolation
  - Multicore support
  - Temporal isolation – MCS needs to be completed!
  - Spatial / physical resources
- Inspectable
- Composable, verified API
- Performance must be good enough

OS Requirements

- Operational Requirements
  - Faults and updates must verifiably have minimal runtime impact
  - Projectable UI/OT integration
- Administrative requirements
  - Manage multiple stakeholders
  - Regulatory support and isolation
- Developer requirements
  - Tooling, documentation, languages, distribution, servers, support, etc.
  - Runtime environments – C/Rust/BEAM/VMs/Containers
  - Ecosystem sharing – libraries, baseboard packages, more

The only thing that matters

The Mission

Come find out what we are doing about all of this

- Sign up for evaluation builds at our table
- See working demos