Towards Secure and Timely Exchange of PMU and other Power System Data

Tim Yardley, Erich Heine, Klara Nahrstedt, Rakesh B. Bobba and William H. Sanders

Information Trust Institute

University of Illinois

{yardley, eheine, klara, rbobba, whs}@illinois.edu

NASPI Working Group Meeting

Feb 29 – March 1 2012

Orlando, FL

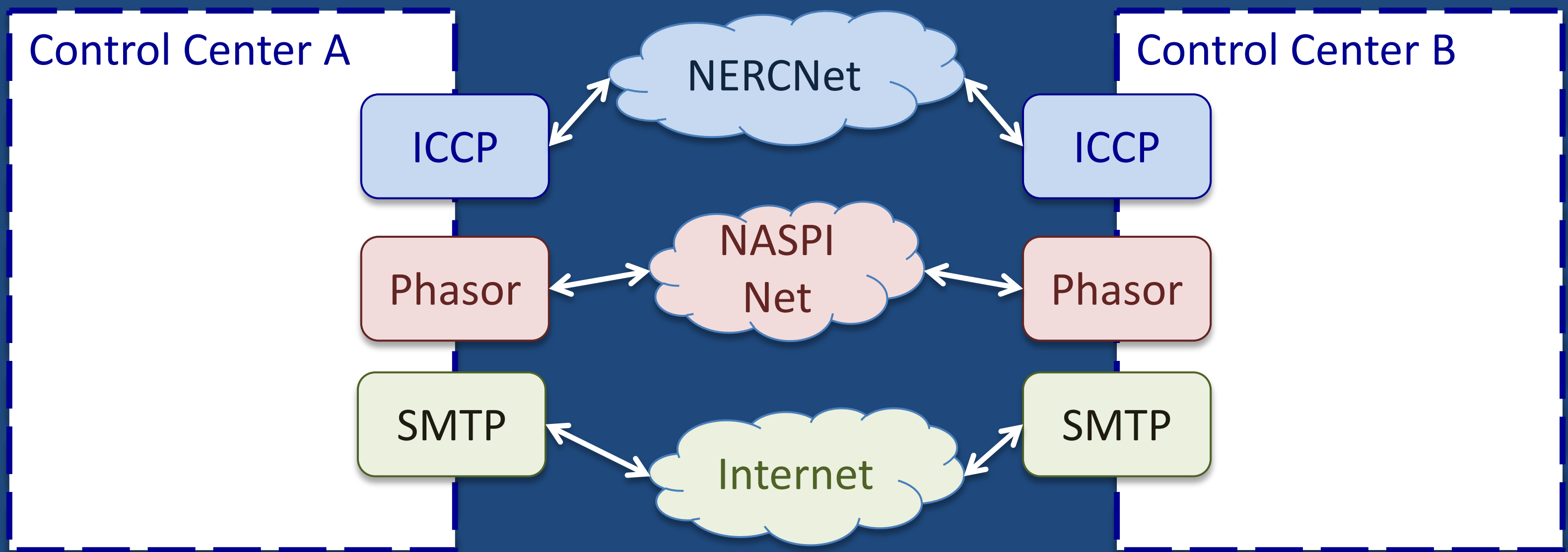
POWER GRID DATA

Data Exchange in Support of Power Systems

- ICCP
- SCADA data
- File-based data (e.g., SDX, Interchange summaries)
- Synchrophasor data
- Metadata
- Topologies
- Compliance or Notifications
- Video and Alarms
- Email and Phone calls
- … and more?

Data Exchange Mechanisms

- ICCP data over NERCNet
- File-based data (e.g., SDX, Interchange summaries)
  - FTP, E-Mail
- PMU data over NASPInet



Looking Ahead

- Data sharing might increase to enable and support new applications
- More types of data may emerge

- Inefficient to maintain multiple networks and mechanisms to exchange different types of data
- Need to "converge" data exchange networks and mechanisms

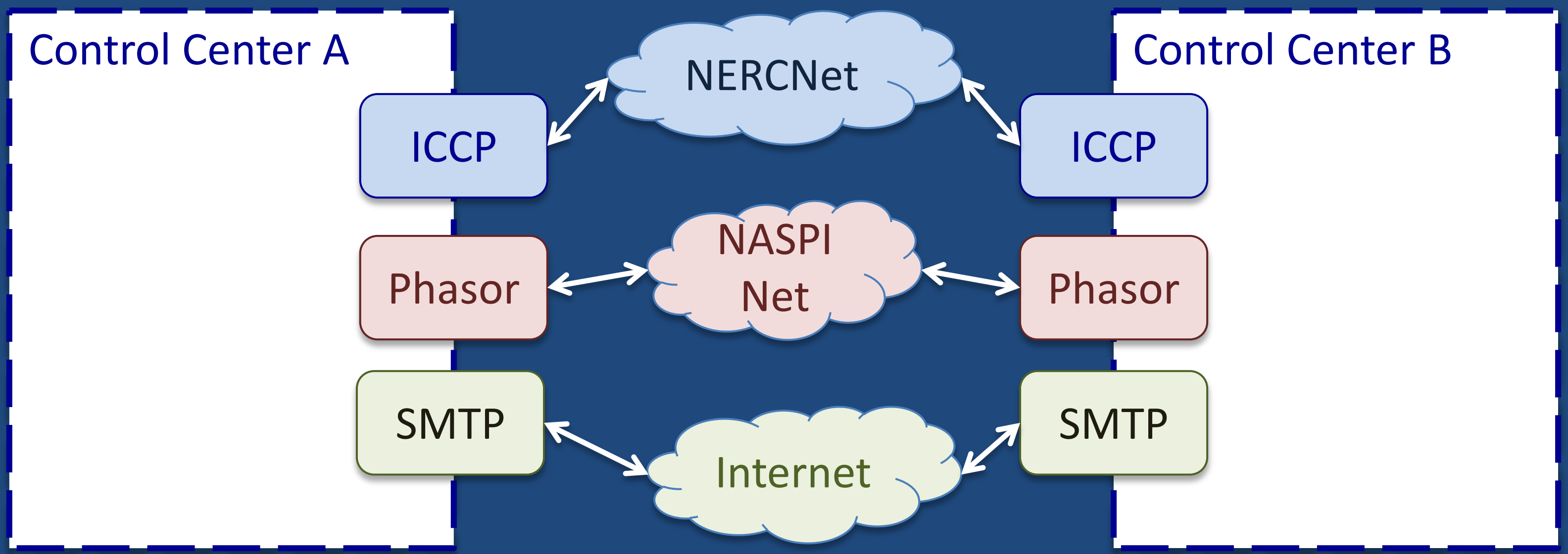
Data Exchange Characteristics & Requirements

- ICCP data over NERCNet
  - Real-time data and control commands
  - Both periodic and event-driven
  - Low latency, high-priority
- File-based data (e.g., SDX, Interchange summaries)
  - Bulk data
  - Periodic
  - Not real-time
- PMU data over NASPInet
  - High sampling rate
  - Multiple data classes
    - Real-time and non-real time (historic)
    - Varying latency requirements (low to high)

THE PROBLEM

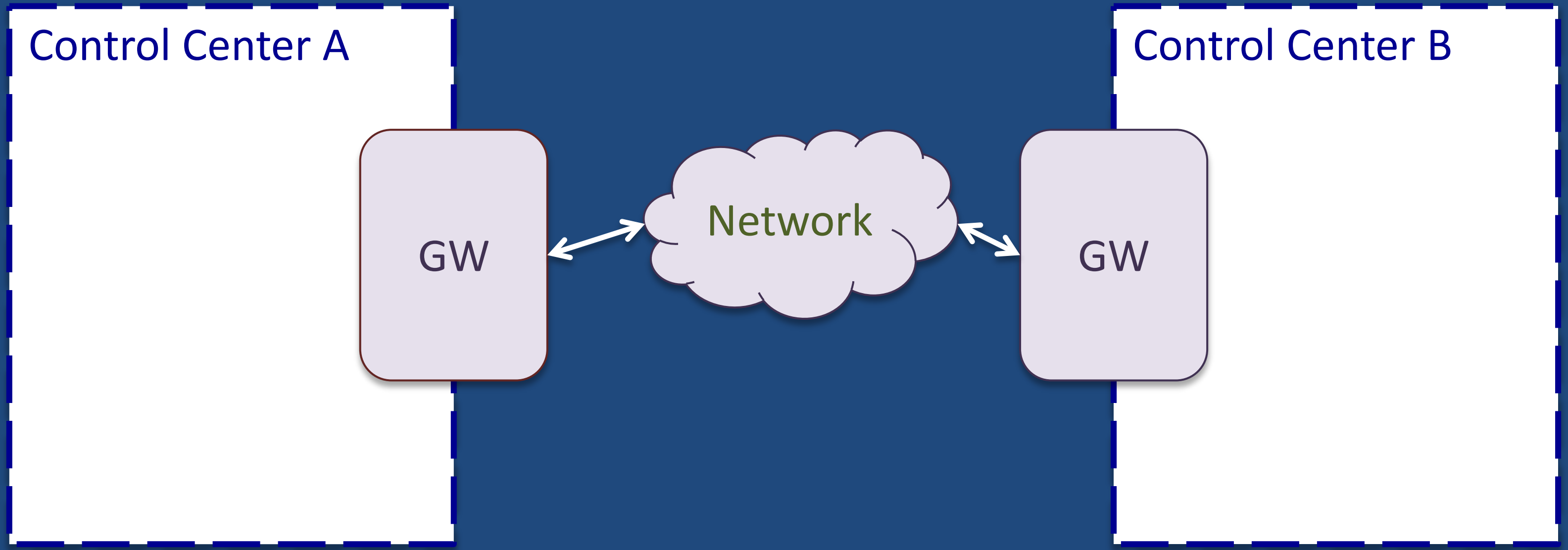
Research Problem

- Objective: Enable data exchange convergence for Power Grid applications
  - Multiple traffic paradigms
  - Understand and support communications requirements/properties for existing and emerging applications



Research Problem

- Objective: Enable data exchange convergence for Power Grid applications
  - Multiple traffic paradigms
  - Understand and support communications requirements/properties for existing and emerging applications



Research Challenges

- Resource management both at the host and on the network
  - Quality of Service, Real-time scheduling, Wide area network optimization
- Security
  - Access control, Integrity, Availability and Confidentiality

Research Challenges

Our focus is on the host

- Resource management both at the host and on the network
  - Quality of Service, Real-time scheduling, Wide area network optimization
- Security
  - Access control, Integrity, Availability and Confidentiality

- In the past we worked on convergence for substation to control center networks – CONES, a DOE funded project

THE WORK

Development and Integration challenges

- Use commercial, off-the-shelf platforms and tools
- Make it easy for utilities to integrate
- Support legacy devices and applications
- Support existing and emerging applications

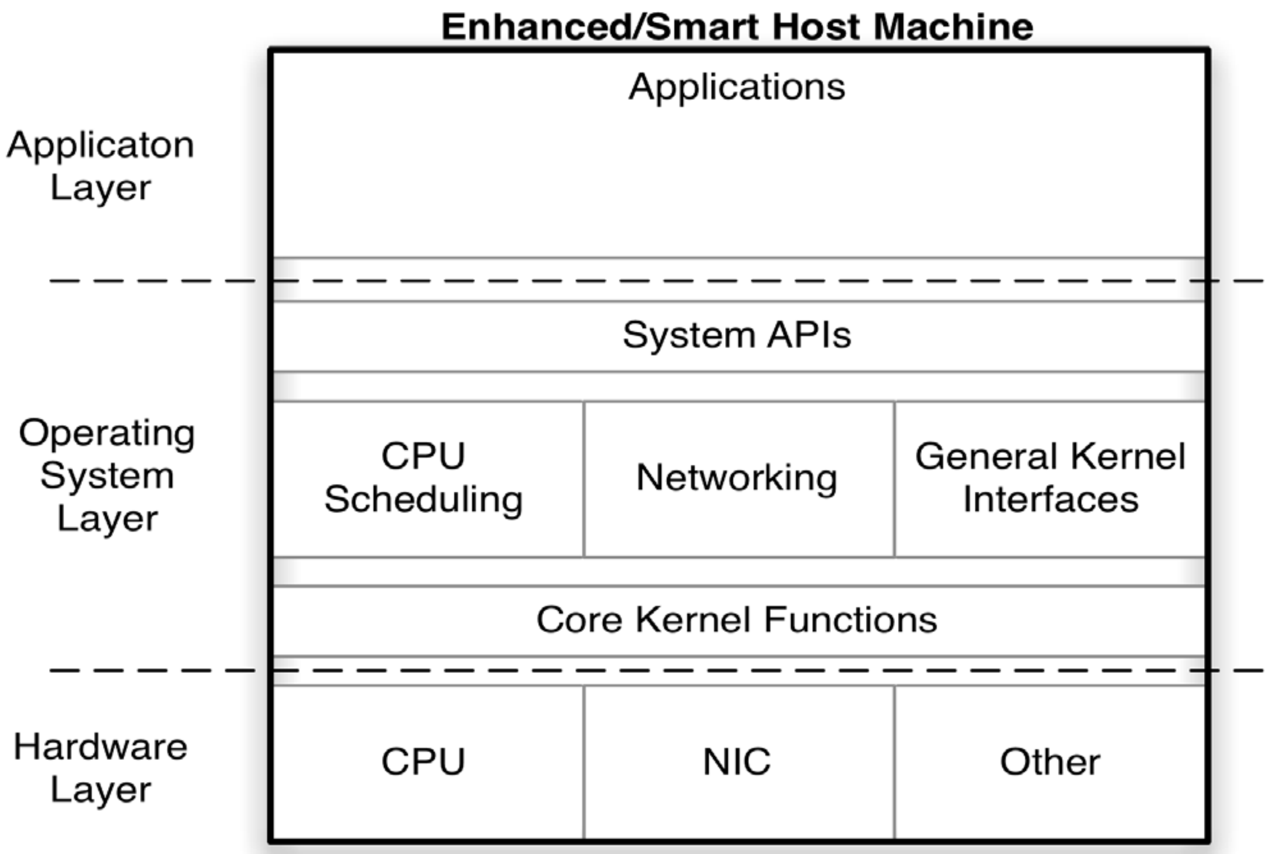
- Provide value and benefit over existing solutions

Architecture and Design Approach

- Real-time: support tight timing guarantees
  - CPU scheduling
  - Network scheduling

- Middleware: abstracting system specifics
  - Common and unified API
  - Coordinate multiple resource management
  - Dynamic reprioritization of QoS

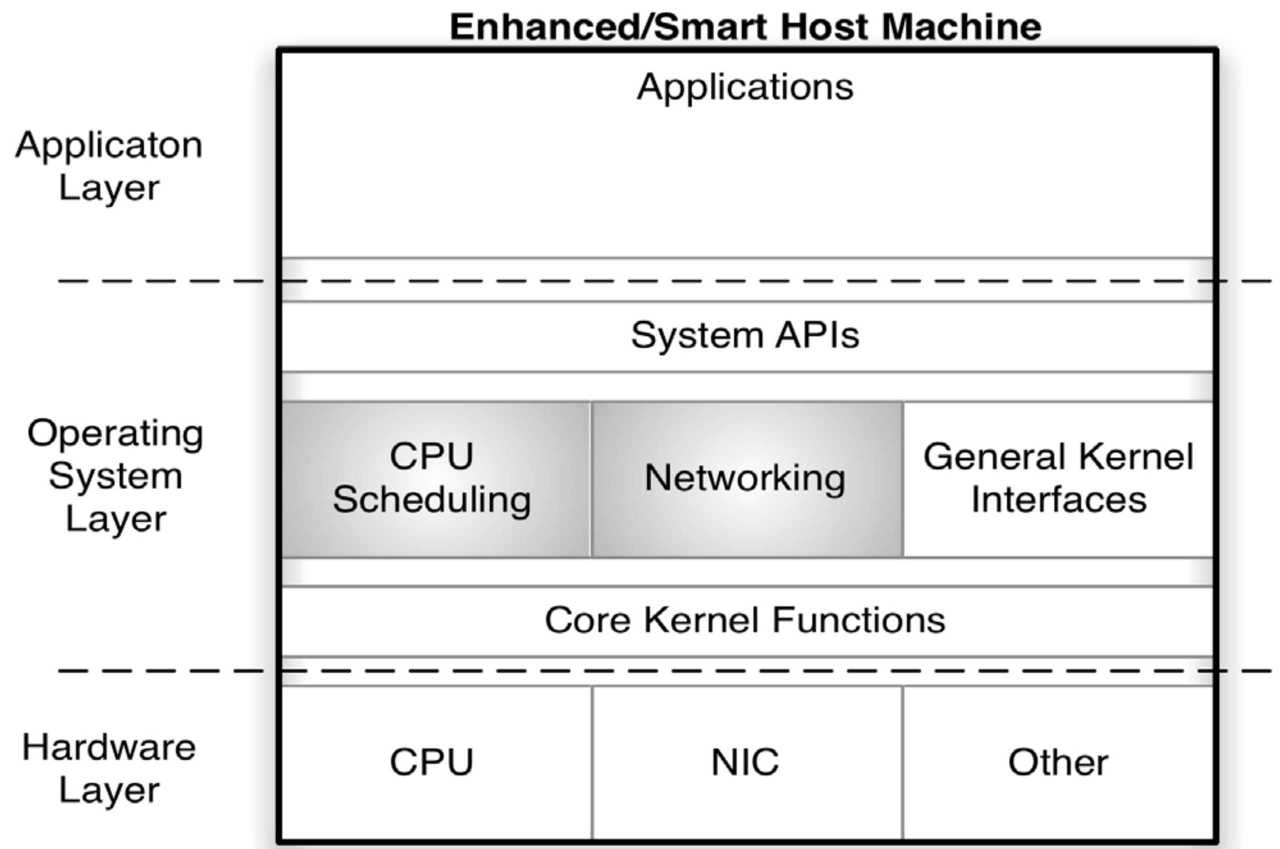
Development – starting point



Base operating system:

- Linux OS
- Preemption and other RT extensions
- iDSRT RT enhancements
- Standard system APIs and userland APIs

Development – kernel enhancement

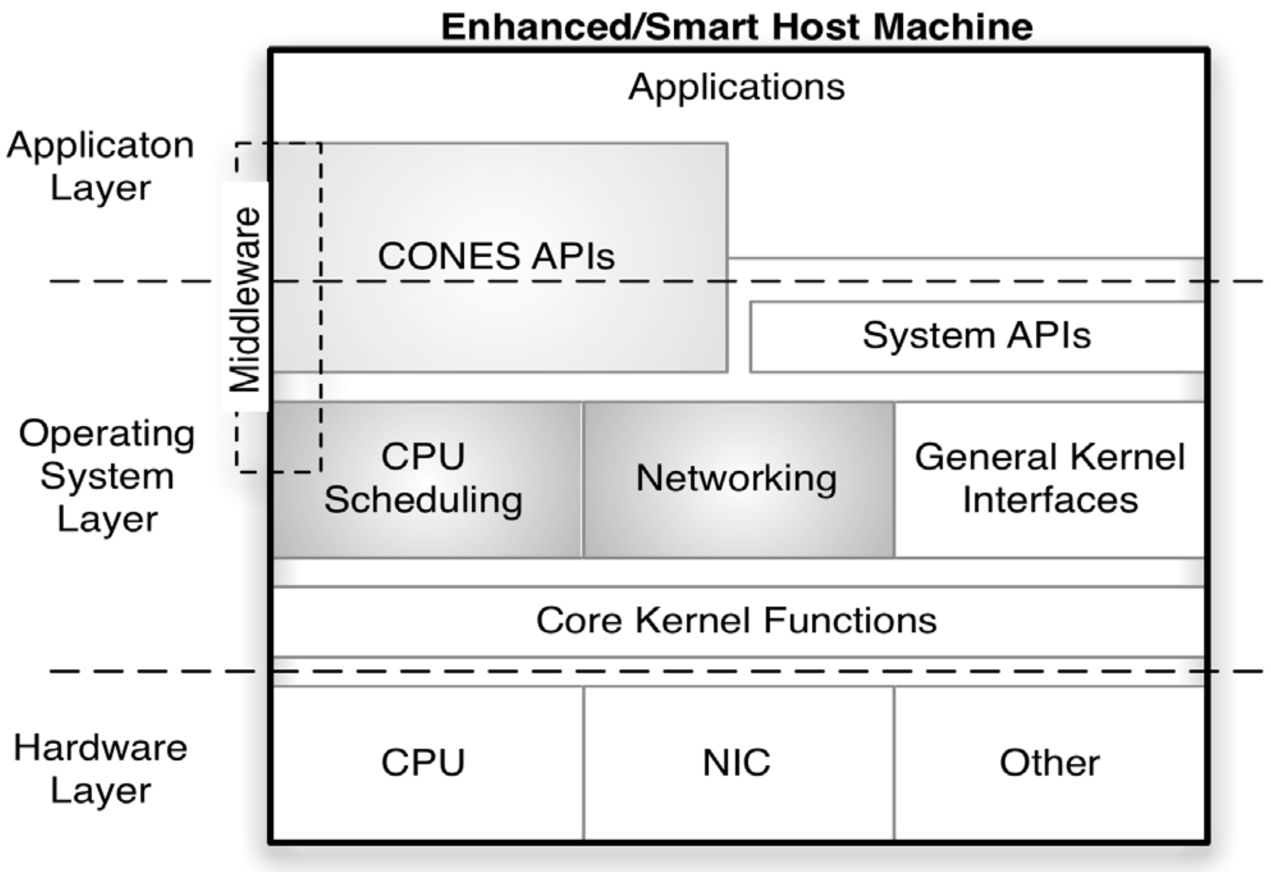


Scheduling Modification

CPU:
- EDF scheduler
- arbitrary process support
- run-time parameter modification

Networking:
- Add EDF based packet scheduler
- Create SO_REALTIME socket option for simple integration.
- Runtime flexibility in packet scheduling.

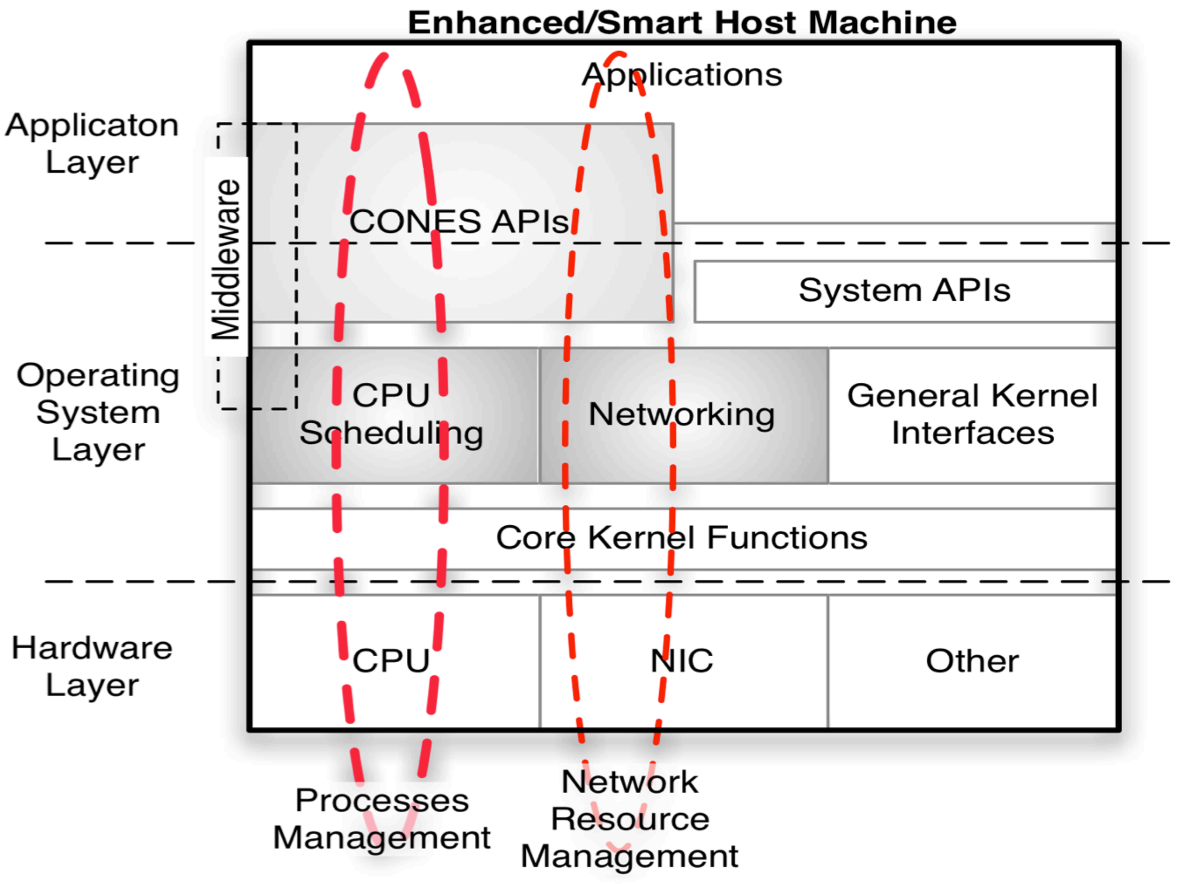
Development - middleware



Middleware Layer:

- Unified API for easy use
- Extensible with coordination hooks
- Portable to other systems
- Allows for simplified integration.

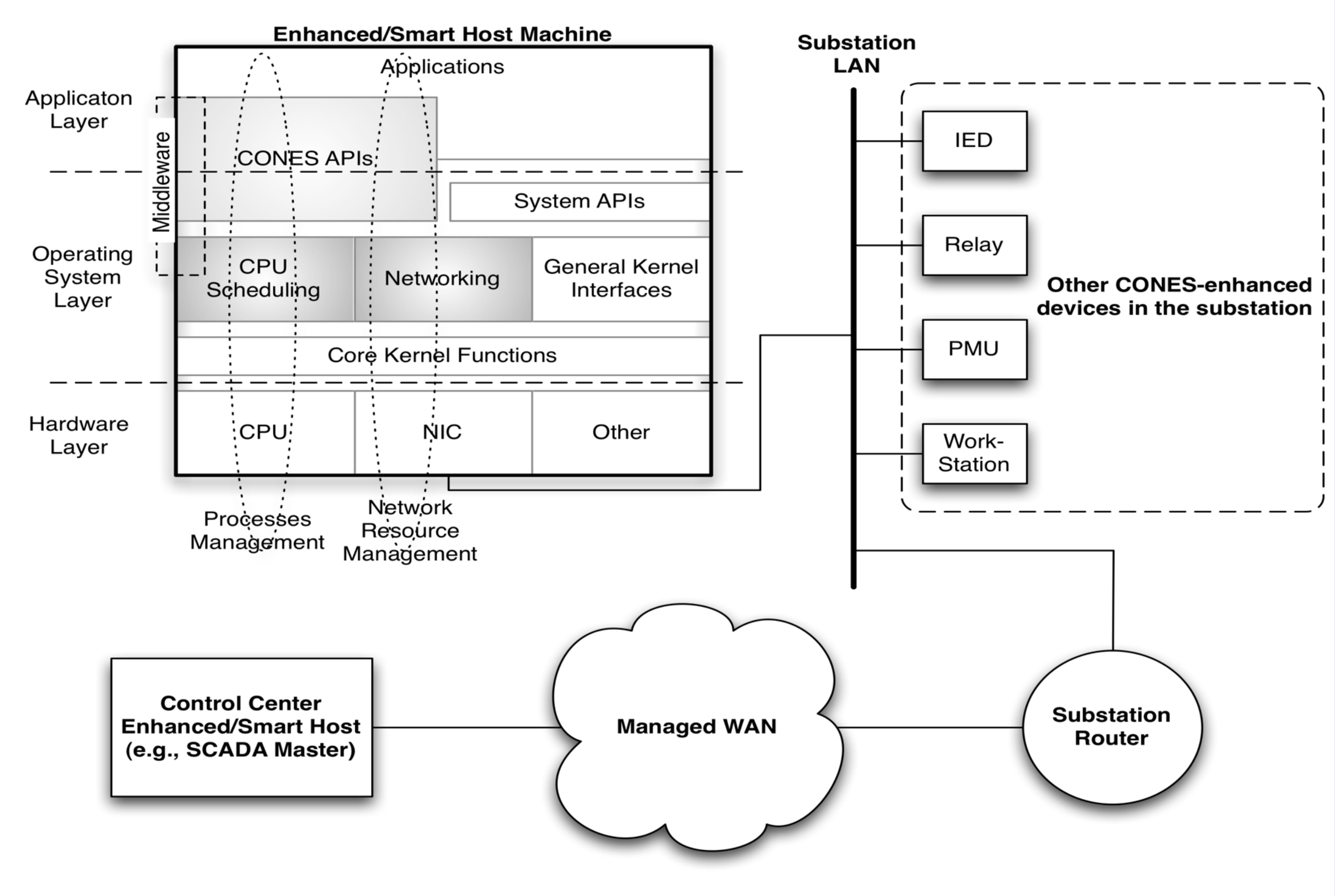
Development – full stack



Full stack effects:

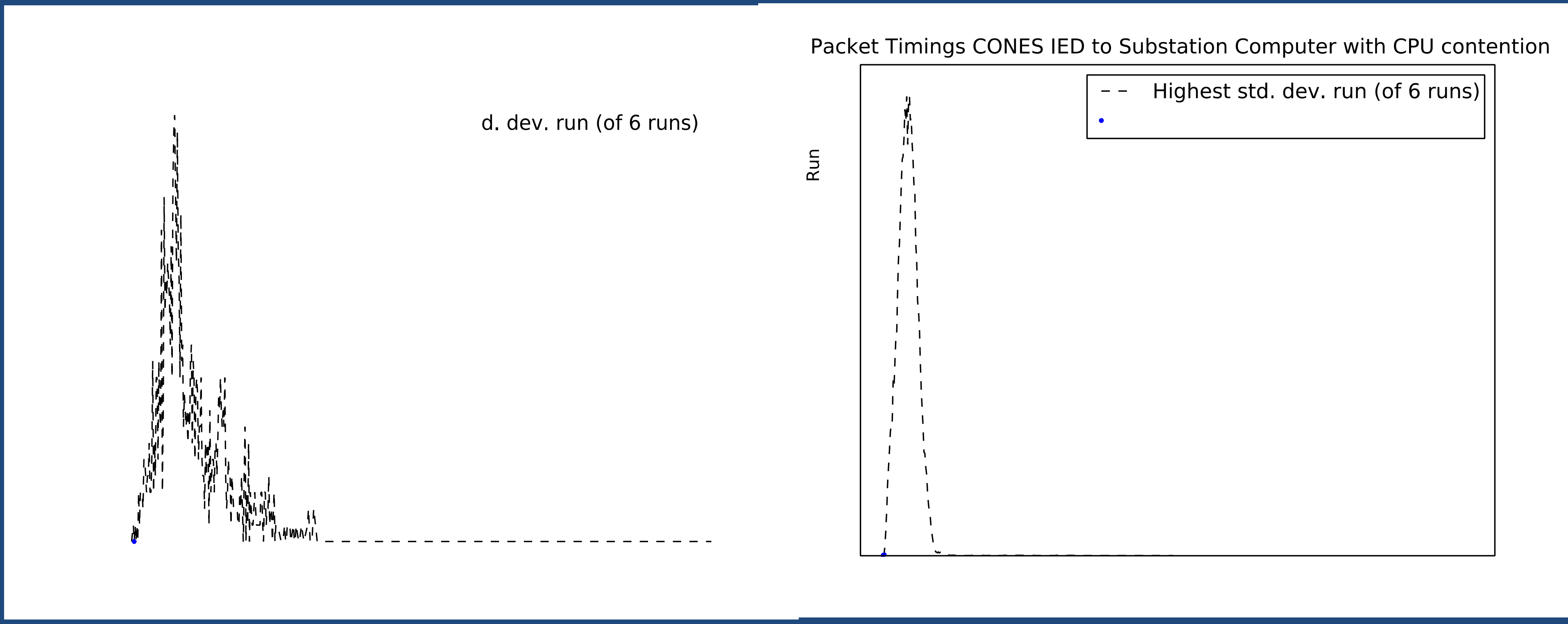
- Top to bottom realtime (soft) guarantees
- Full system resource management
- Processes which are somewhat immune to 3rd party programming errors.

Wide Area Architecture

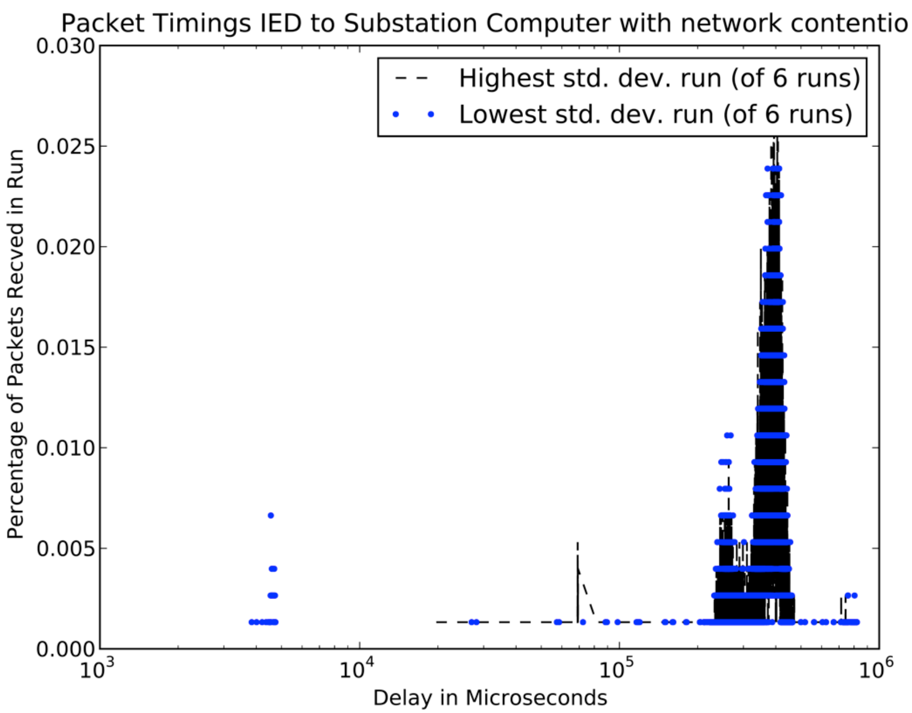


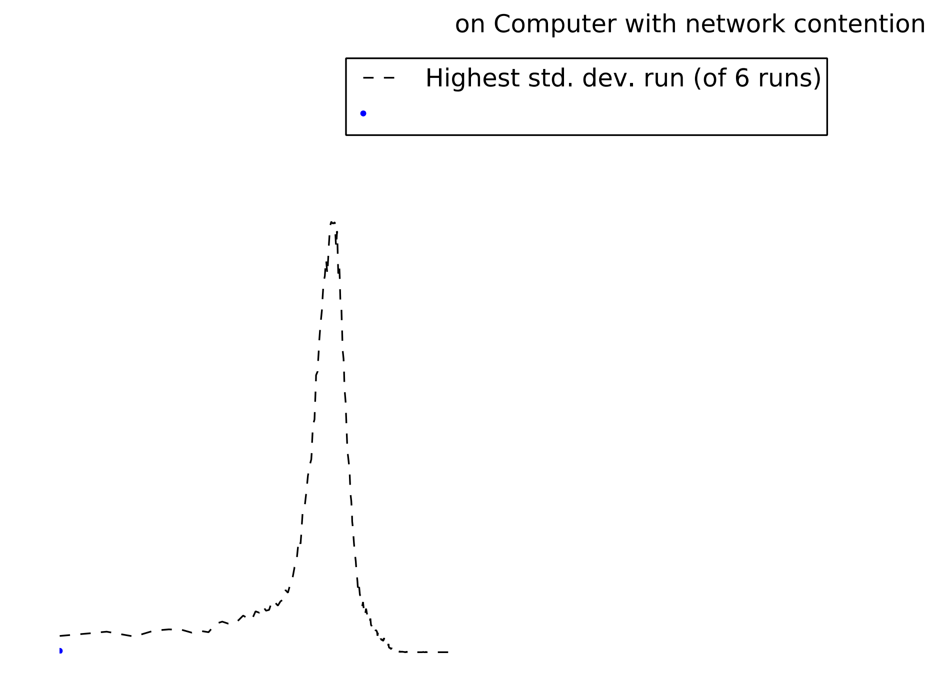
RESULTS

Results – CPU contention



Results – network contention





Lessons Learned

- Convergence is achievable
  - But it needs to be architected carefully

- Realtime augmentation of CPU and Network provide tremendous benefit when faced with contention and competition for resources
  - But, this requires kernel and potentially driver modification on most Oss

- When control isn't involved, the problem is a bit simpler.

THE FUTURE

Applying Lessons Learned

- Inter Control Center Data Exchanges including mixed-mode hosts
- Operating constraints when adapting from Linux and applying on Windows

- Also being applied and commercialized via the DOE sponsored project SIEGate, jointly developed by GPA and UIUC
  - Participants: GPA (prime), UIUC, PNNL, PJM, and Alstom Grid

SIEGate

- Handles more power grid data types (as discussed)
- Leverages the code base of OpenPDC and OpenPG
- Contains intelligence for differentiating streams of data to achieve QoS constraints
- Incorporates and leverages lessons learned from CONES to build a more scalable and adaptable system
- Provides a consistent security footprint for externally communicating power grid data

Summary

- Cost effective, flexible communications needed in the future.
- Timing constraints must be met at the endpoints and in the "middle".
- We have built the basis for a solid platform, but the lessons learned are what are most important.

See http://tcipg.org for more

QUESTIONS?