Grid Protection Alliance Update

NASPI Working Group Meeting

October 5-6, 2010

Arlington, VA

GRID PROTECTION ALLIANCE

Primary Initiatives Status

- PDC Test Bench
  - Beta version available
- PMU Registry
  - Accepting data entry on web site
  - Distributed system deployment developing
- Phasor Gateway
  - Moving towards NASPInet

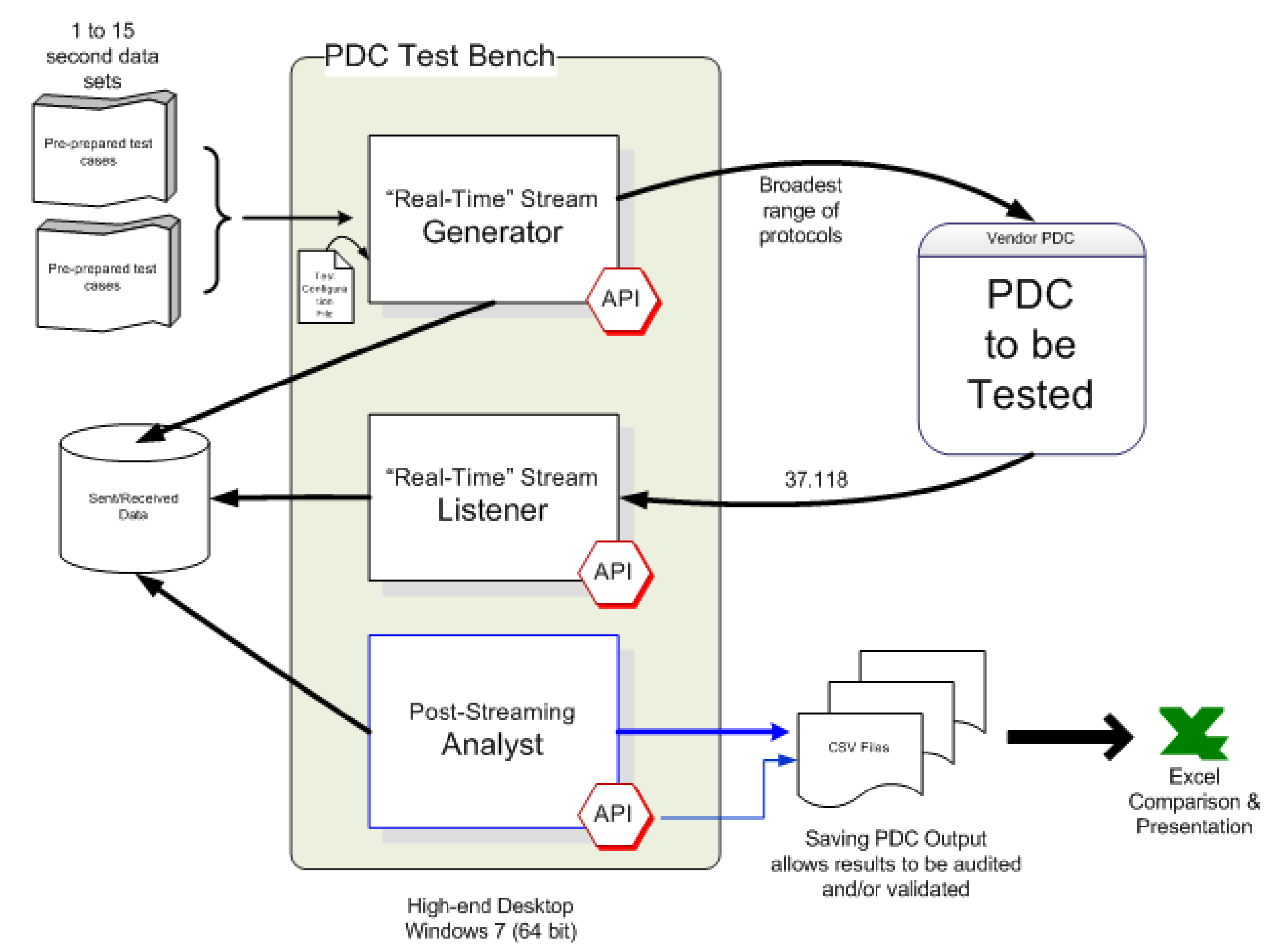
PDC Test Bench Purpose

The purpose of the PDC Test Bench is to test a Phasor Data Concentrator (PDC) by varying the inputs and comparing the subsequent outputs with expected results.

Project Requirements

- Must support the common phasor protocols
- Must be able to vary input
- Must be able to produce more data than can be accepted by any PDC
- Must produce meaningful and reproducible results
- Must be configurable by the end user
- Must support extended periods of testing
- Must support analysis export in human readable comma separated files
- Must support custom analysis modules

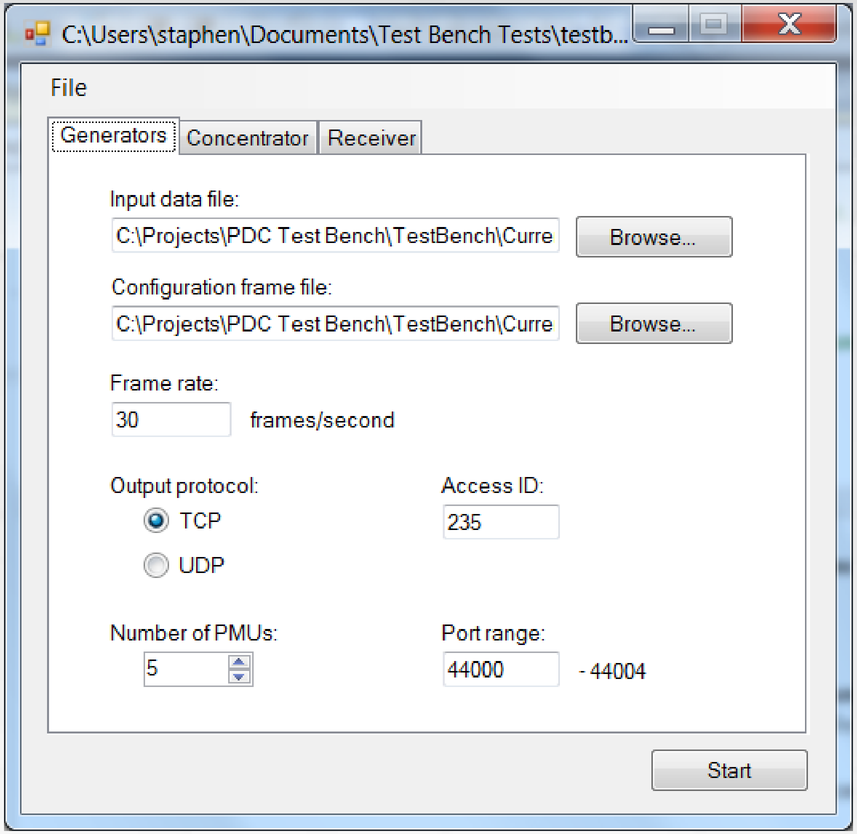
PDC Test Bench Overview


1 to 15 second data sets
Pre-prepared test cases
Pre-prepared test cases
PDC Test Bench
"Real-Time" Stream Generator
Test Configuration File
API
Broadest range of protocols
Vendor PDC
PDC to be Tested
Sent/Received Data
"Real-Time" Stream Listener
API
37.118
Post-Streaming Analyst
API
CSV Files
Excel Comparison & Presentation
Saving PDC Output allows results to be audited and/or validated
High-end Desktop Windows 7 (64 bit)

Using the PDC Test Bench

- Configure and start PDC.
- Enter information into all fields on the Test Bench interface.
- Save changes to a package file.
- Start the PDC Test Bench. Note: The concentrator must be running before starting the Test Bench.
- When satisfied, stop the PDC Test Bench. Test results are stored in the receiver’s output file and timestamp file.

Test Bench Images



Input data file:
C:\Projects\PDC Test Bench\TestBench\Curre

Configuration frame file:
C:\Projects\PDC Test Bench\TestBench\Curre

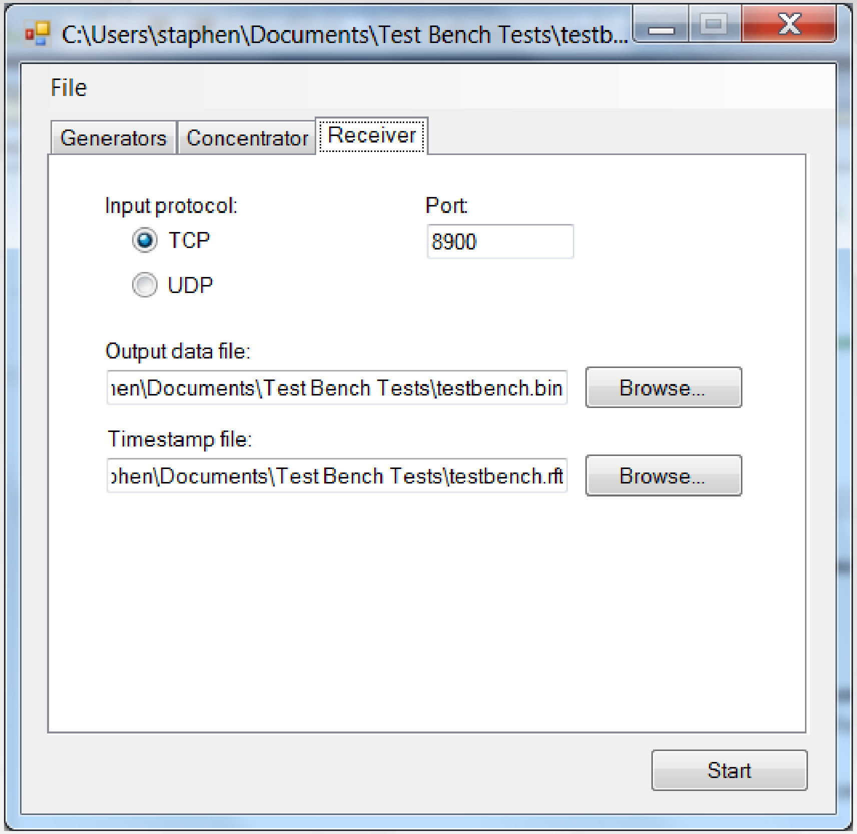
Frame rate:
30 frames/second

Output protocol:
● TCP
○ UDP

Access ID:
235

Number of PMUs:
5

Port range:
44000 - 44004



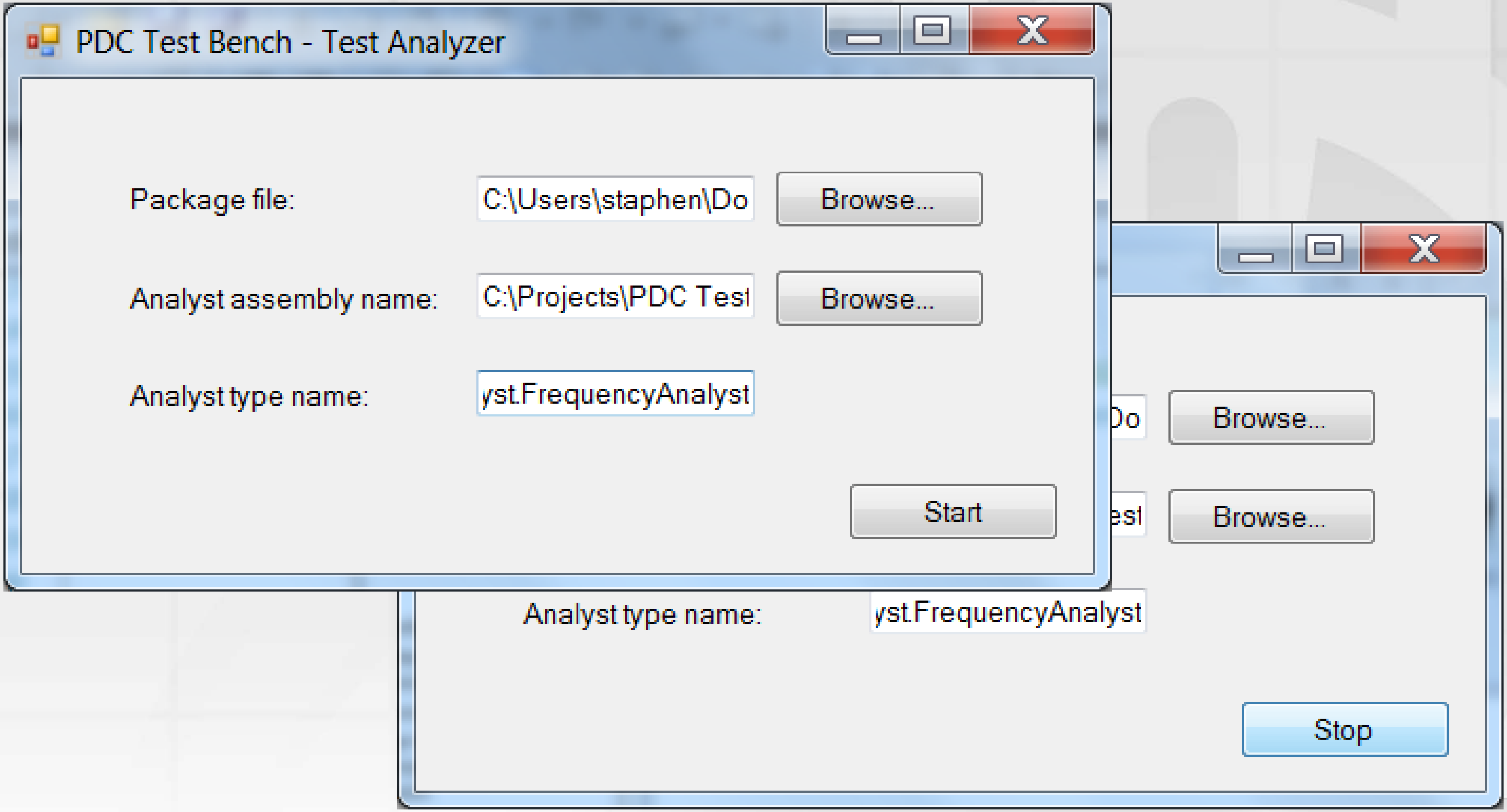
Input protocol:
● TCP
○ UDP

Port:
8900

Output data file:
ıen\Documents\Test Bench Tests\testbench.bin

Timestamp file:
ɔhen\Documents\Test Bench Tests\testbench.rft

Analyzer Images



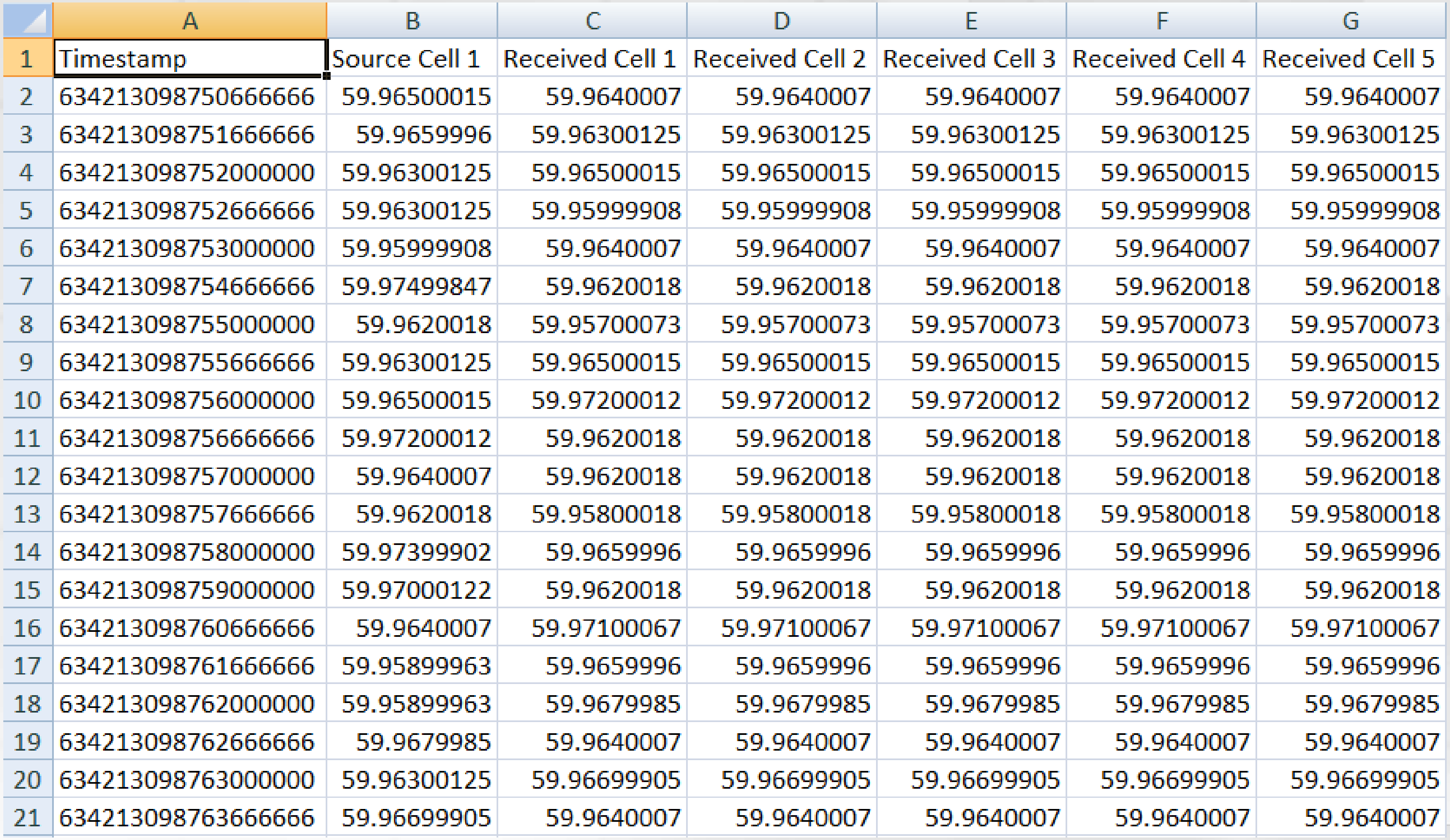
PDC Test Bench - Test Analyzer

Package file: C:\Users\staphen\Do

Analyst assembly name: C:\Projects\PDC Test

Analyst type name: yst.FrequencyAnalyst

Analyst Example Results



	A	B	C	D	E	F	G
1	Timestamp	Source Cell 1	Received Cell 1	Received Cell 2	Received Cell 3	Received Cell 4	Received Cell 5
2	634213098750666666	59.96500015	59.9640007	59.9640007	59.9640007	59.9640007	59.9640007
3	634213098751666666	59.9659996	59.96300125	59.96300125	59.96300125	59.96300125	59.96300125
4	634213098752000000	59.96300125	59.96500015	59.96500015	59.96500015	59.96500015	59.96500015
5	634213098752666666	59.96300125	59.95999908	59.95999908	59.95999908	59.95999908	59.95999908
6	634213098753000000	59.95999908	59.9640007	59.9640007	59.9640007	59.9640007	59.9640007
7	634213098754666666	59.97499847	59.9620018	59.9620018	59.9620018	59.9620018	59.9620018
8	634213098755000000	59.9620018	59.95700073	59.95700073	59.95700073	59.95700073	59.95700073
9	634213098755666666	59.96300125	59.96500015	59.96500015	59.96500015	59.96500015	59.96500015
10	634213098756000000	59.96500015	59.97200012	59.97200012	59.97200012	59.97200012	59.97200012
11	634213098756666666	59.97200012	59.9620018	59.9620018	59.9620018	59.9620018	59.9620018
12	634213098757000000	59.9640007	59.9620018	59.9620018	59.9620018	59.9620018	59.9620018
13	634213098757666666	59.9620018	59.95800018	59.95800018	59.95800018	59.95800018	59.95800018
14	634213098758000000	59.97399902	59.9659996	59.9659996	59.9659996	59.9659996	59.9659996
15	634213098759000000	59.97000122	59.9620018	59.9620018	59.9620018	59.9620018	59.9620018
16	634213098760666666	59.9640007	59.97100067	59.97100067	59.97100067	59.97100067	59.97100067
17	634213098761666666	59.95899963	59.9659996	59.9659996	59.9659996	59.9659996	59.9659996
18	634213098762000000	59.95899963	59.9679985	59.9679985	59.9679985	59.9679985	59.9679985
19	634213098762666666	59.9679985	59.9640007	59.9640007	59.9640007	59.9640007	59.9640007
20	634213098763000000	59.96300125	59.96699905	59.96699905	59.96699905	59.96699905	59.96699905
21	634213098763666666	59.96699905	59.9640007	59.9640007	59.9640007	59.9640007	59.9640007

GRID
PROTECTION
ALLIANCE

What’s Available in the BETA?

- Full framework for performing detailed stream analysis of cached output
- Example test analyzer that computes simple latency (received time minus sent time) and compares input and output frequencies
- Extendible to perform more detailed tests

this is work for you!

GRID PROTECTION ALLIANCE

Where do I get it?

http://pdctestbench.codeplex.com/

Warning:

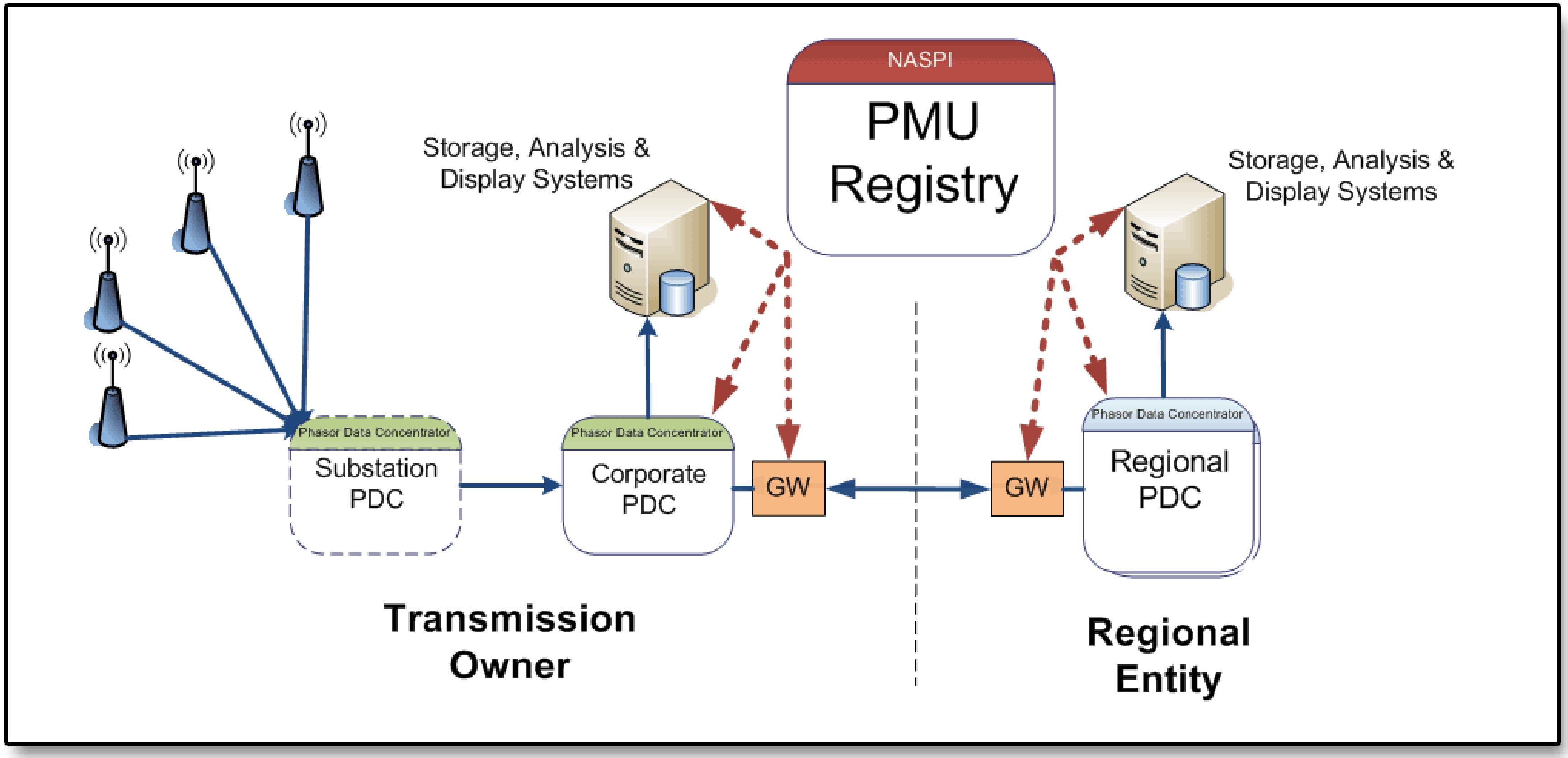
Some ~~assembly~~ coding required. ☺

What is the PMU Registry?

The NASPI PMU Registry is the source of meta data on synchrophasor devices and the measurements collected by them throughout North America.

- Where is the measurement taken?
- What is the measurement?
- Who owns the measurement?
- … and for gateways, where can I get it?

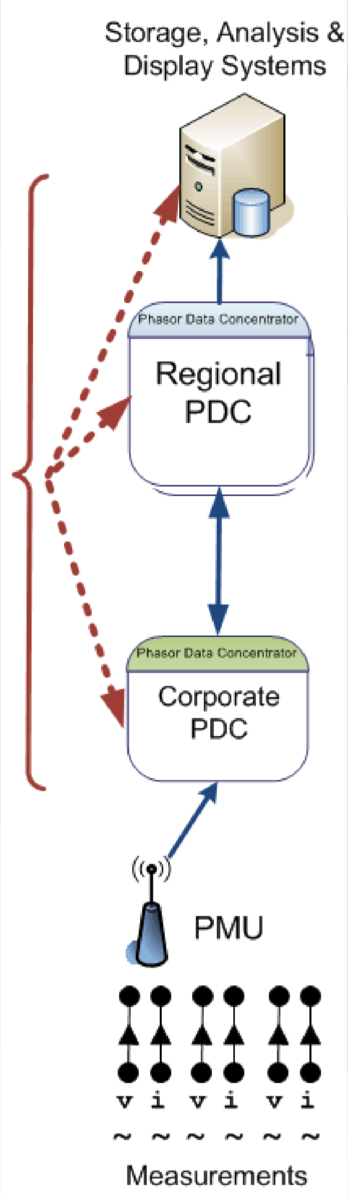
Generic Architecture



GRID PROTECTION ALLIANCE

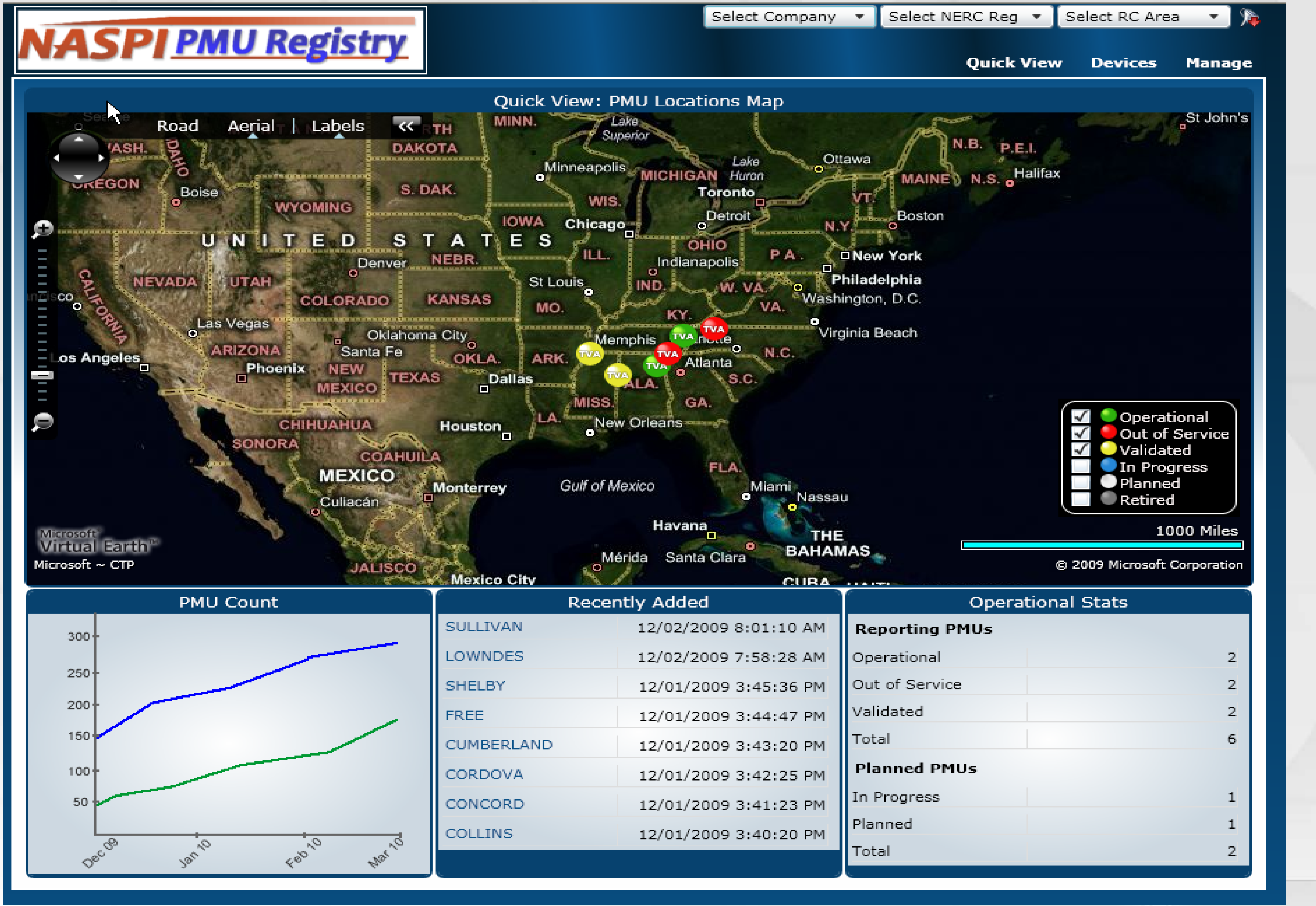
PMU Registry Data Overview

- Simple Data Hierarchy
  - Regional concentration
  - Corporate (TO) concentration
  - PMU
  - Measurement Descriptors
- Allows for multiple layers of data concentration and creation of “virtual measurements”
- Allows PMUs that do not provide data to external users to be a part of the registry
- Creates standard IDs and stores common names for PMUs as well as measurement descriptors.



The PMU Registry only holds meta data.

What’s here today? Metadata…



Quick View: PMU Locations Map

☑ 🟢 Operational
☑ 🔴 Out of Service
☑ 🟡 Validated
□ 🔵 In Progress
□ ⚪ Planned
□ ⚫ Retired

Recently Added

SULLIVAN	12/02/2009 8:01:10 AM
LOWNDES	12/02/2009 7:58:28 AM
SHELBY	12/01/2009 3:45:36 PM
FREE	12/01/2009 3:44:47 PM
CUMBERLAND	12/01/2009 3:43:20 PM
CORDOVA	12/01/2009 3:42:25 PM
CONCORD	12/01/2009 3:41:23 PM
COLLINS	12/01/2009 3:40:20 PM

Operational Stats

Reporting PMUs	
Operational	2
Out of Service	2
Validated	2
Total	6
Planned PMUs	
In Progress	1
Planned	1
Total	2

GRID PROTECTION ALLIANCE

Beta version released in June 2010

PMU Registry Plans

- SGIG project winners currently investigating registry requirements for their individual architectures
- Clear need to accelerate development of components to allow deployment as an integrated, multi-node system
- PMU Registry established as it’s own open source project in August 2010
- Security and framework (.net 4.0) improvements planned for January 2011 release.

How do I get to it?

https://www.naspi.net/

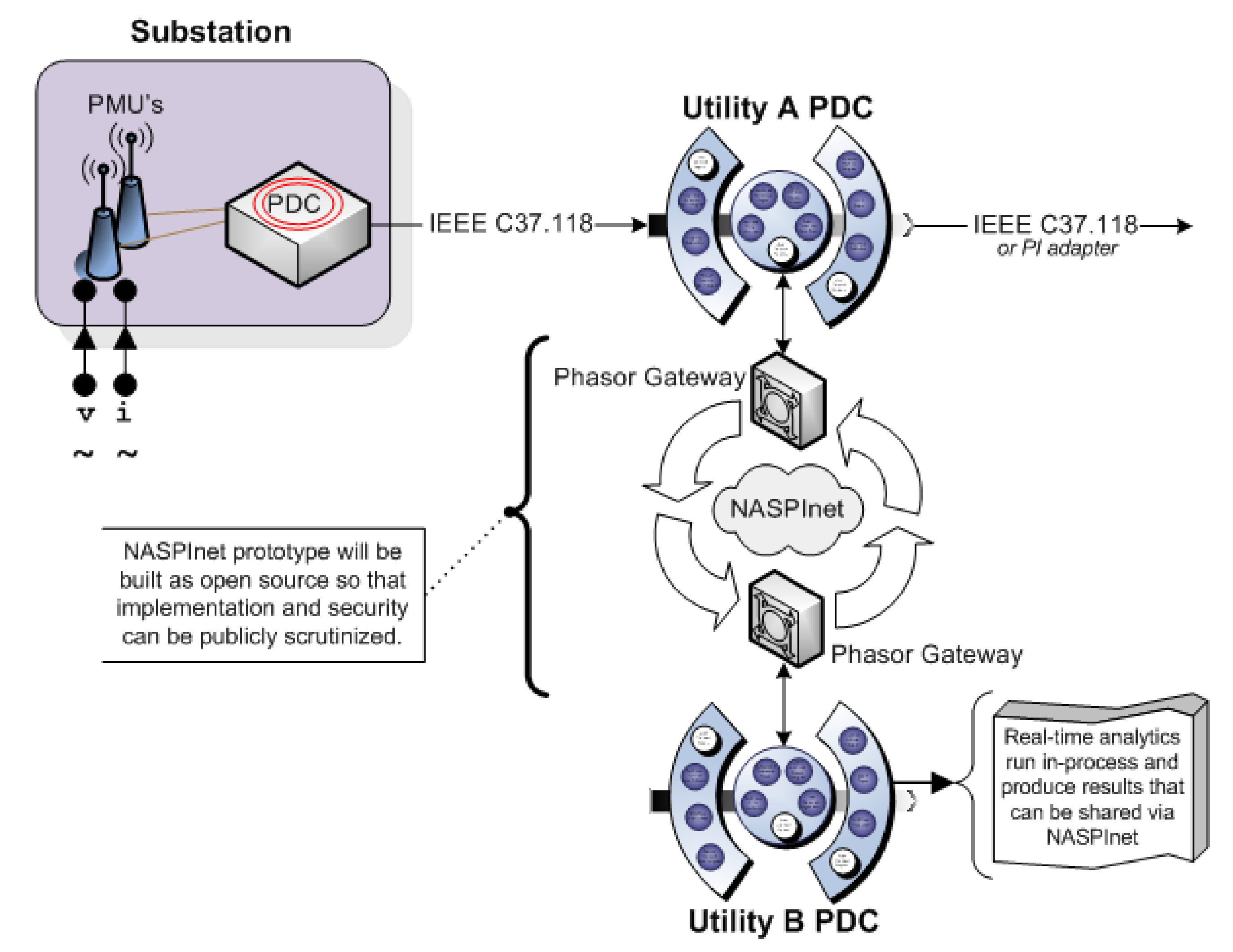
Source code also available:

http://pmuregistry.codeplex.com/

Setting the stage for NASPInet

- To publish a signal on NASPInet, an entity must:
  - Have a NASPInet Phasor Gateway
  - Register the measurement (signal) with the NASPInet infrastructure.
  - Configure the Phasor Gateway to designate the authorized receiving Gateways (this configuration can be “all”) for this signal.
- To subscribe to a signal on NASPInet, an entity must:
  - Have a NASPInet Phasor Gateway
  - Discover the measurement (signal) needed through the NASPInet infrastructure
  - Request authorization to receive this signal from the publishing Phasor Gateway owner.

First Step: An Open Synchrophasor Gateway

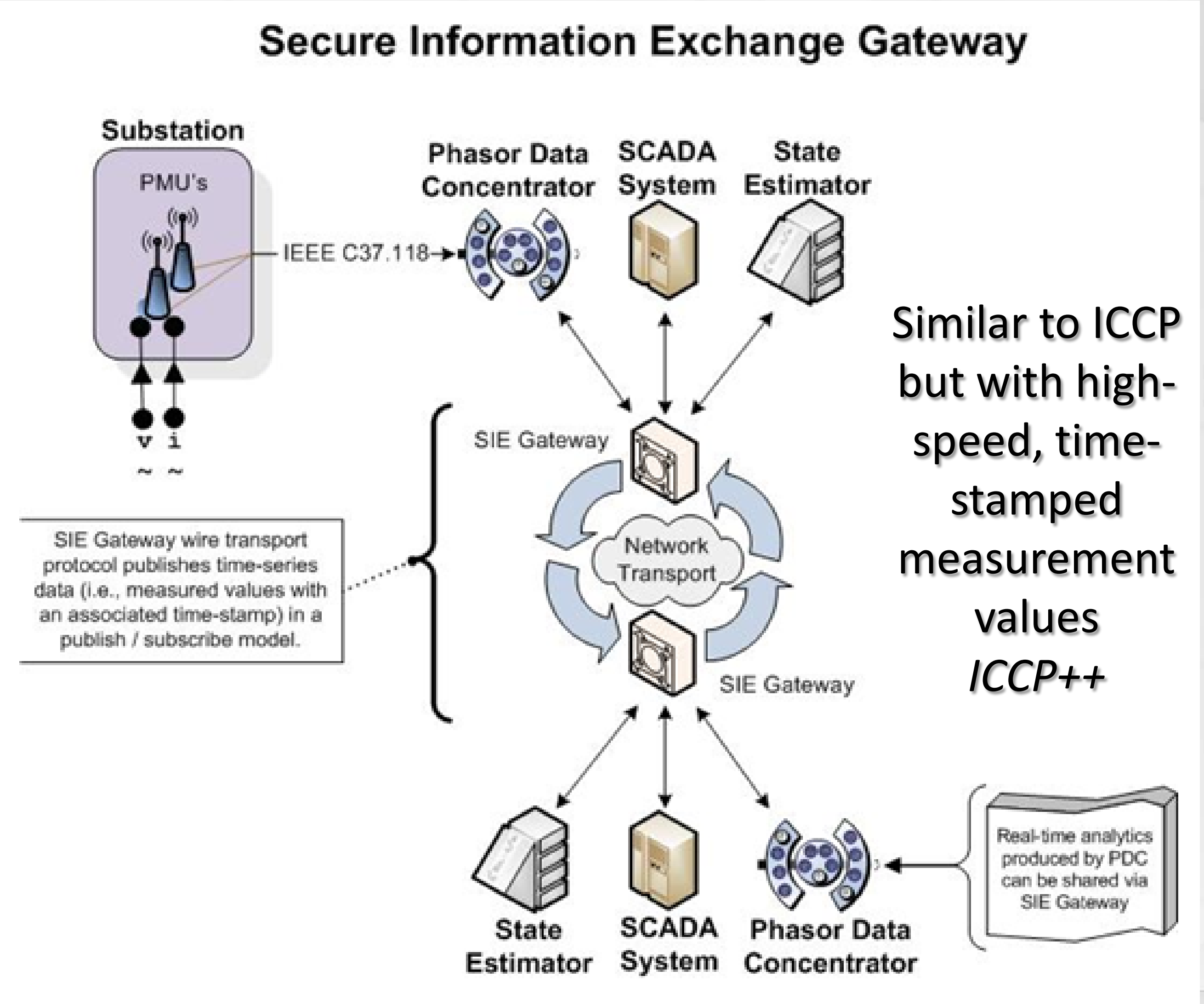


NASPInet prototype will be built as open source so that implementation and security can be publicly scrutinized.

Real-time analytics run in-process and produce results that can be shared via NASPInet

GRID PROTECTION ALLIANCE

End Goal: A General Purpose Data Gateway



SIE Gateway wire transport protocol publishes time-series data (i.e., measured values with an associated time-stamp) in a publish / subscribe model.

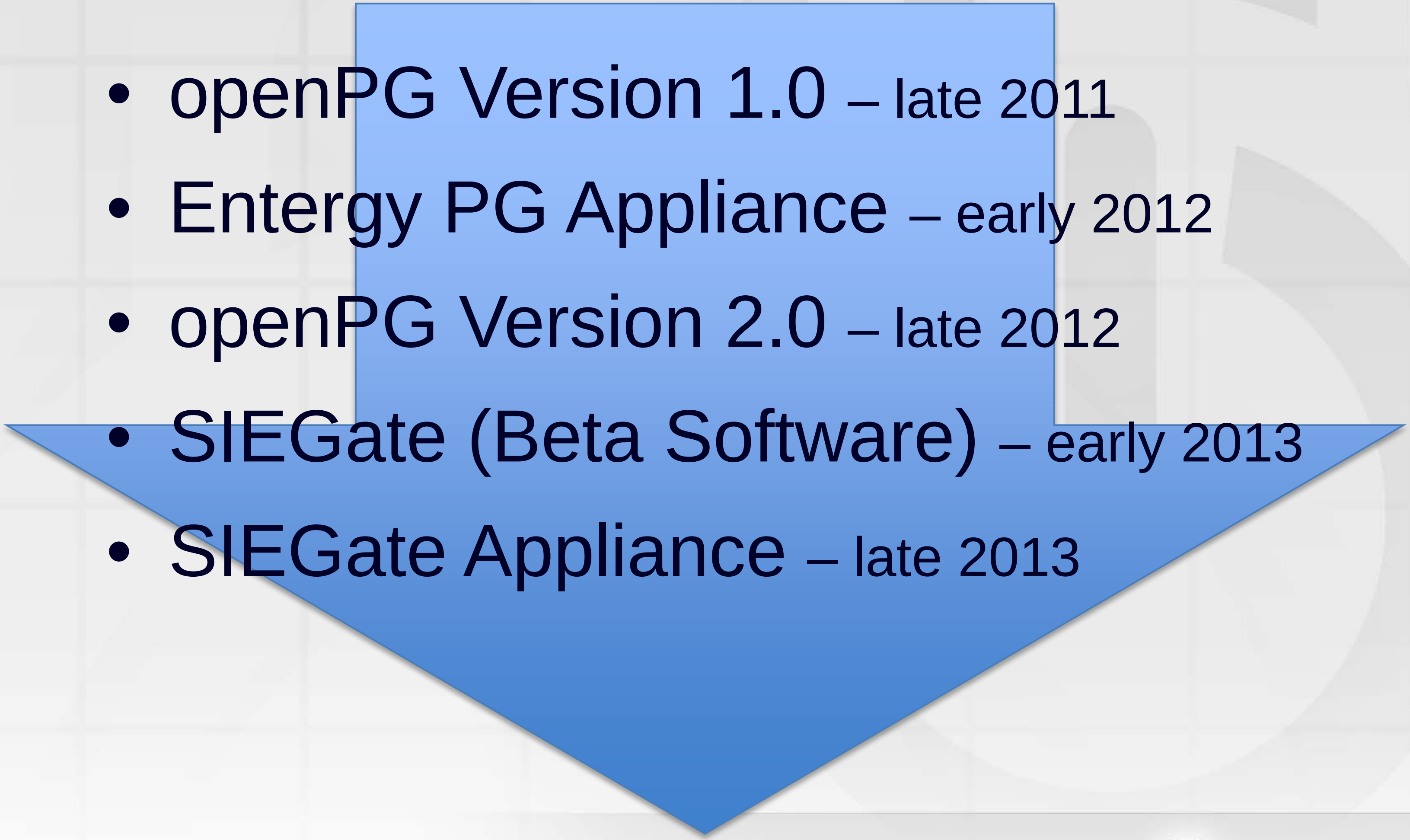
Real-time analytics produced by PDC can be shared via SIE Gateway

Similar to ICCP but with high-speed, time-stamped measurement values

ICCP++

GRID PROTECTION ALLIANCE

Phasor Gateway Plan Summary



- openPG Version 1.0 – late 2011
- Entergy PG Appliance – early 2012
- openPG Version 2.0 – late 2012
- SIEGate (Beta Software) – early 2013
- SIEGate Appliance – late 2013

The progression toward NASPInet.

SIEGate

A generalized, security hardened appliance for the exchange of real-time grid operating information that is both open source and commercialized through a major vendor

- Available in 2013
- Developed by UIUC/GPA
- Commercialization by ALSTOM
- Testing by PNNL
- Demonstration at ALSTOM and PJM
- Funded by DOE-CEDS with a major cost-share contribution by NERC

GRID PROTECTION ALLIANCE