

# GPA Product Architecture

2012 User's Forum Tutorial

J. Ritchie Carroll

8/21/2012

# Open Source Code Projects

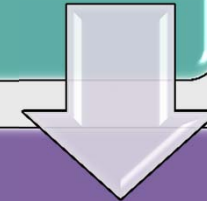
## Code Library

<http://tvacodelibrary.codeplex.com/>



## Time-series Framework (TSF)

<http://timeseriesframework.codeplex.com/>



## TSF Implementations:

openPDC / openPG / openHistorian / SIEGate

<http://openpdc.codeplex.com/>

# Code Library Purpose

- General purpose open source library of .NET code used by the TVA, other utilities and various open source projects that contains a large variety of code useful for nearly any .NET project.
- Consists of hundreds of class libraries that extend and expand the functionality included in the .NET Framework making more complex .NET features easier to use and adds functions not included in the .NET Framework.
- Used since it provides a standard development platform, improves development speed and increases reliability.

# Time-series Framework Purpose

- Core collection of classes used to manage, process and respond to dynamic changes in fast moving streaming time-series data in real-time.
- Allows applications to be architected as measurement routing systems using “Input”, “Action” and “Output” adapter layer. Any application can host the framework which will allow the system become a “measurement router”.

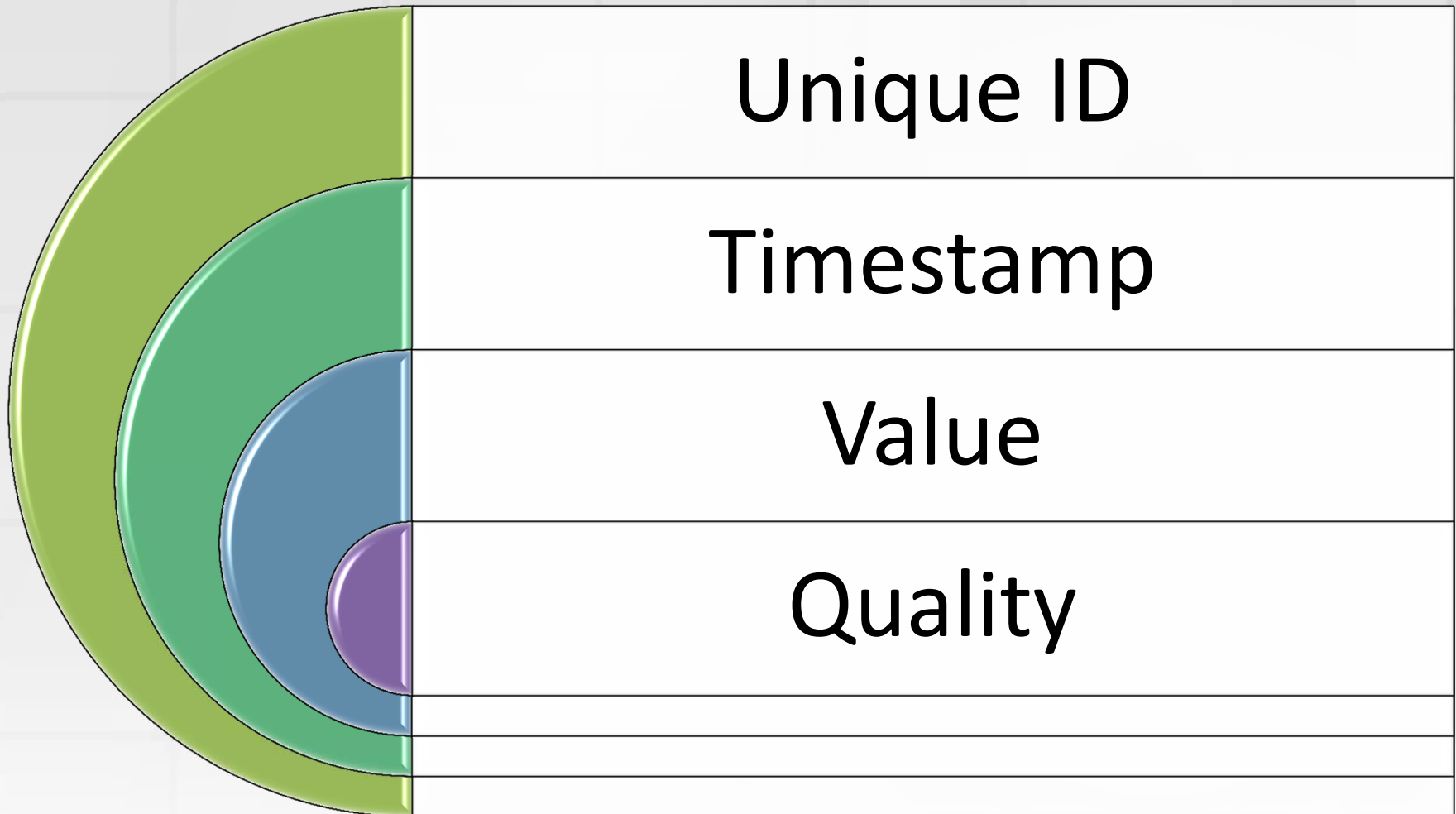
# TSF Implementations

- openPDC
  - <http://openpdc.codeplex.com/>
- openPG
  - <http://openpg.codeplex.com/>
- openHistorian
  - <http://openhistorian.codeplex.com/>
- SIEGate
  - <http://www.siegate.com/>

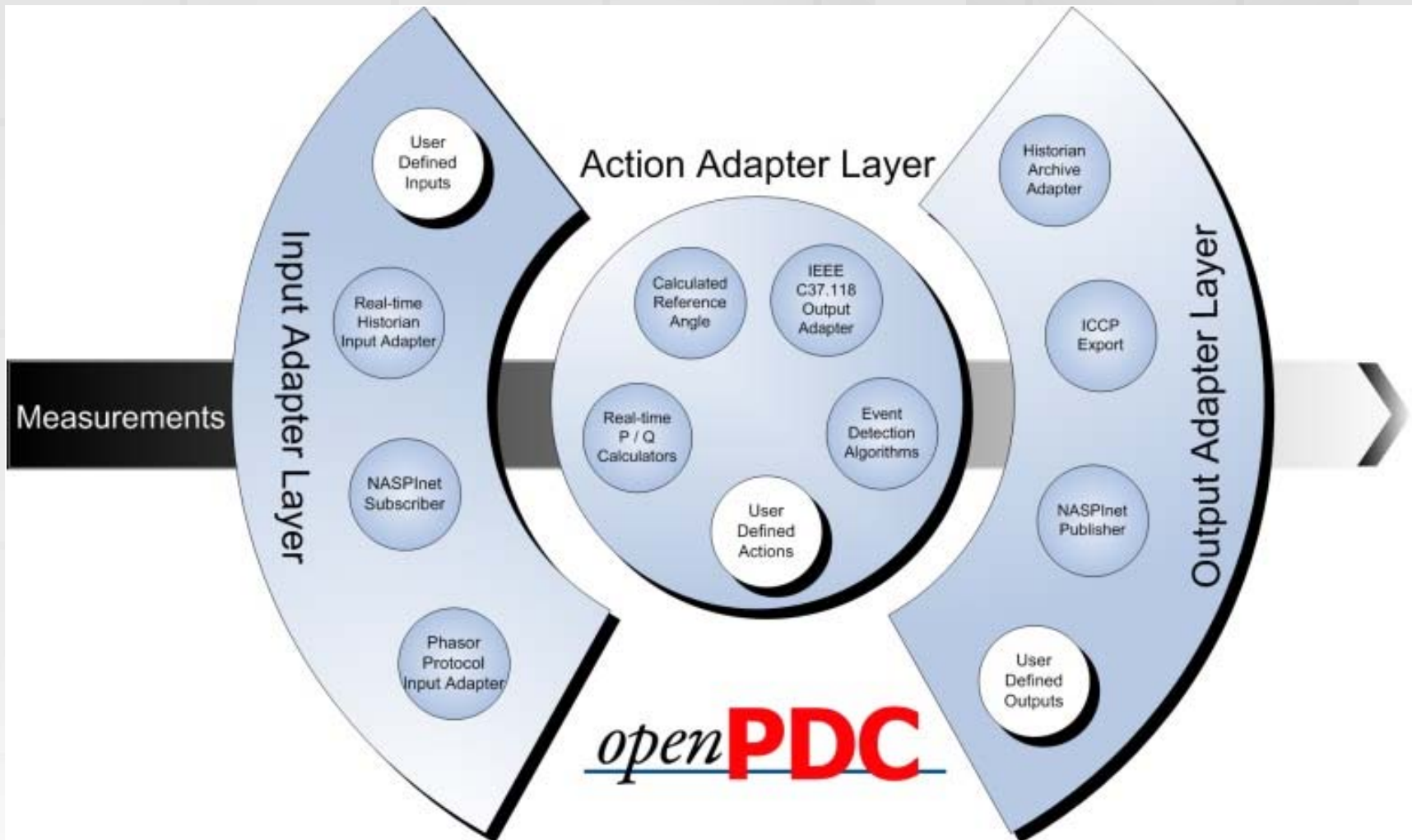
# Measurements

- Numeric quantities that have been acquired at a source device are often known as points, signals, events, or time-series values. Inside TSF they are known as ***measurements***.
- Examples include: temperature, voltage, vibration, location, luminosity and phasors.
- In the synchrophasor world when a value gets measured, an exact timestamp is taken, typically using a GPS-clock for accuracy – the value, along with its timestamp, is then streamed back to TSF host application, e.g., the openPDC, where it can be “time-aligned” with other incoming measurements so that an action can then be taken on a complete slice of data that was all measured at the exact same moment in time.

# Measurement Structure

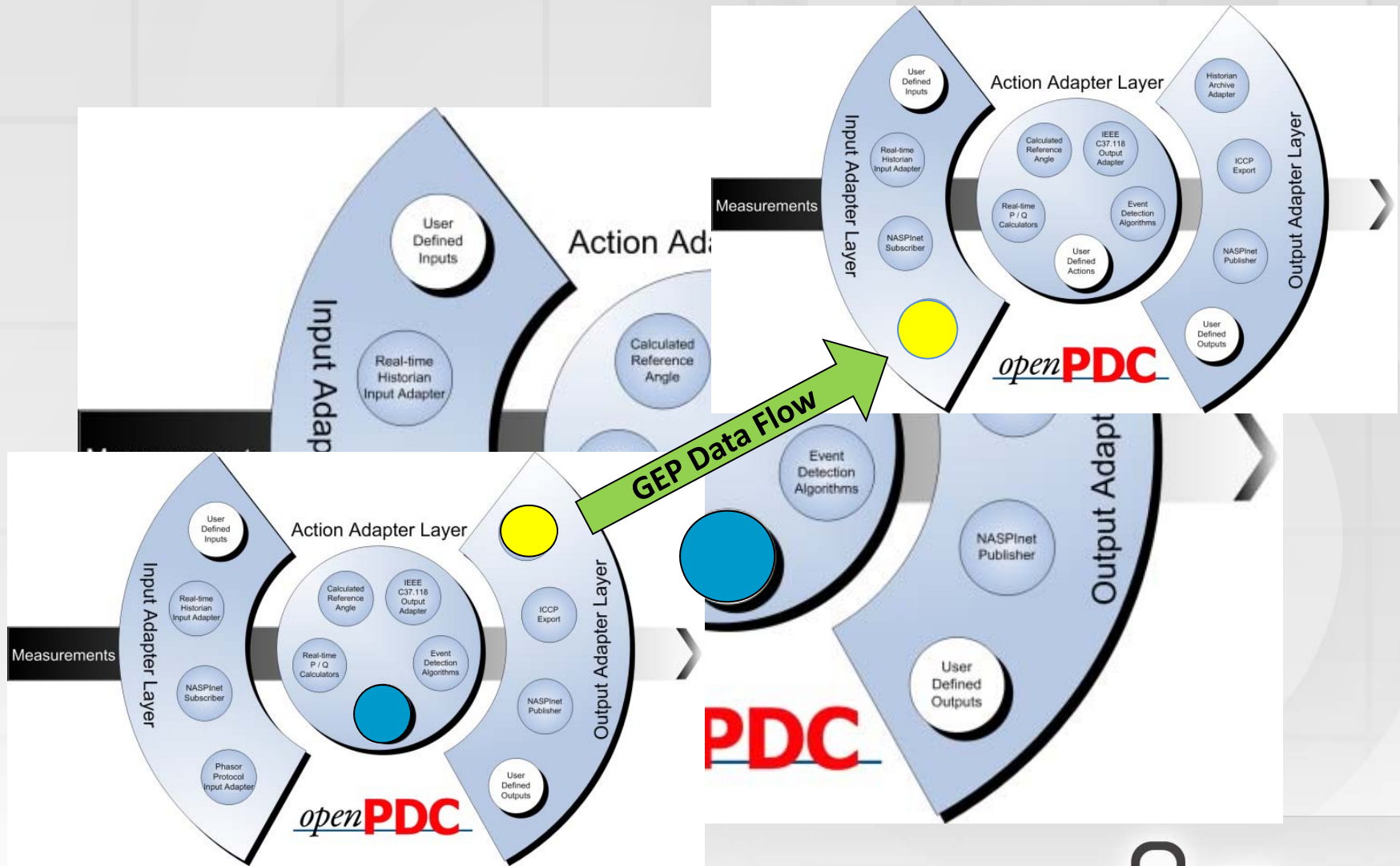


# Adapter Based Architecture





# Scalable Adapter Distribution



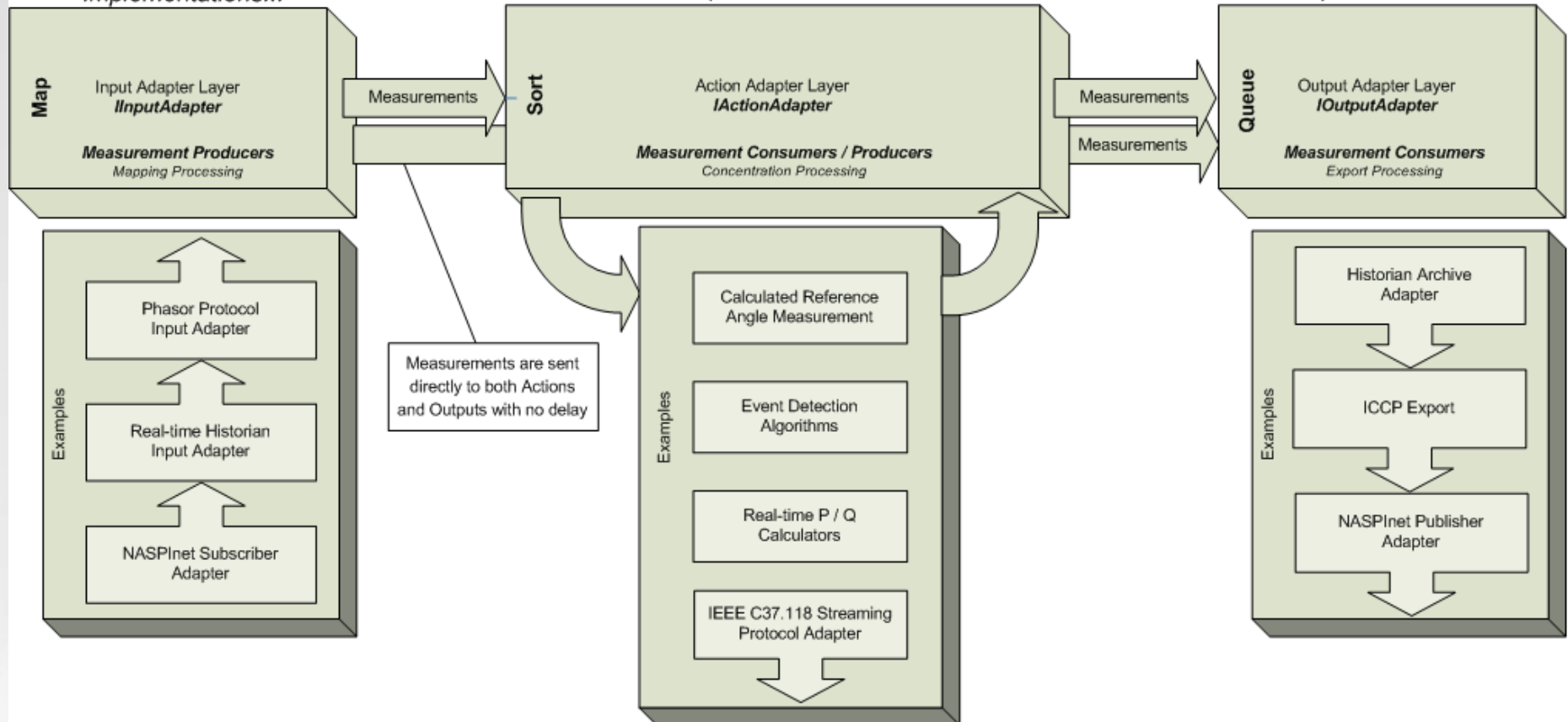
# Input/Action/Output Interface Adapter Layer

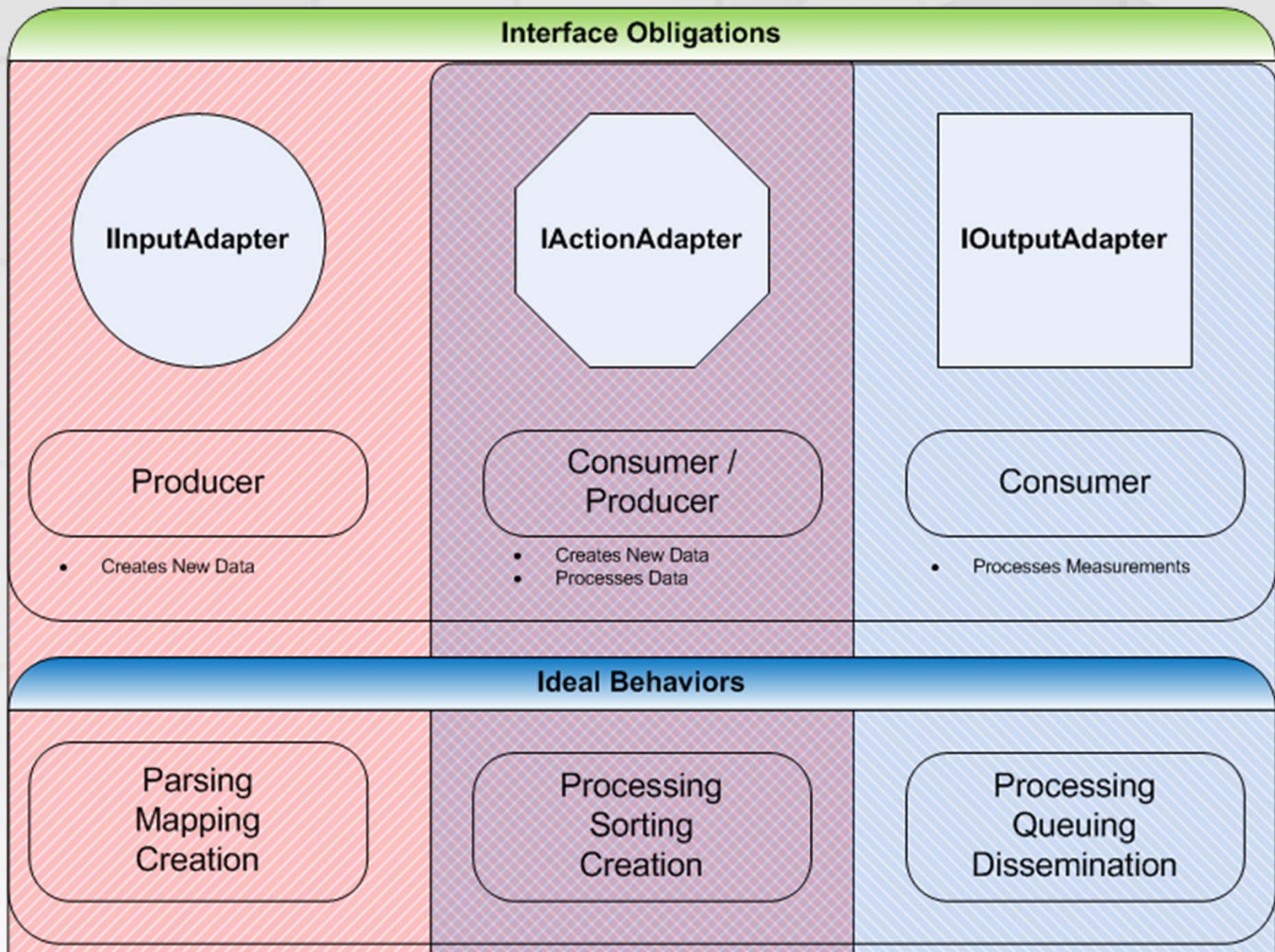
## TimeSeriesFramework.Adapters

Measurements flow into the system via *IInputAdapter* implementations...

Measurements are sorted by time and acted upon by *IActionAdapter* implementations...

Measurements are queued for export by *IOutputAdapter* implementations...





# Input Adapters

*Purpose:*  
**MAP**

- Collect and parse streaming data, assign incoming measurements an ID.

# Output Adapters

*Purpose:*  
**QUEUE**

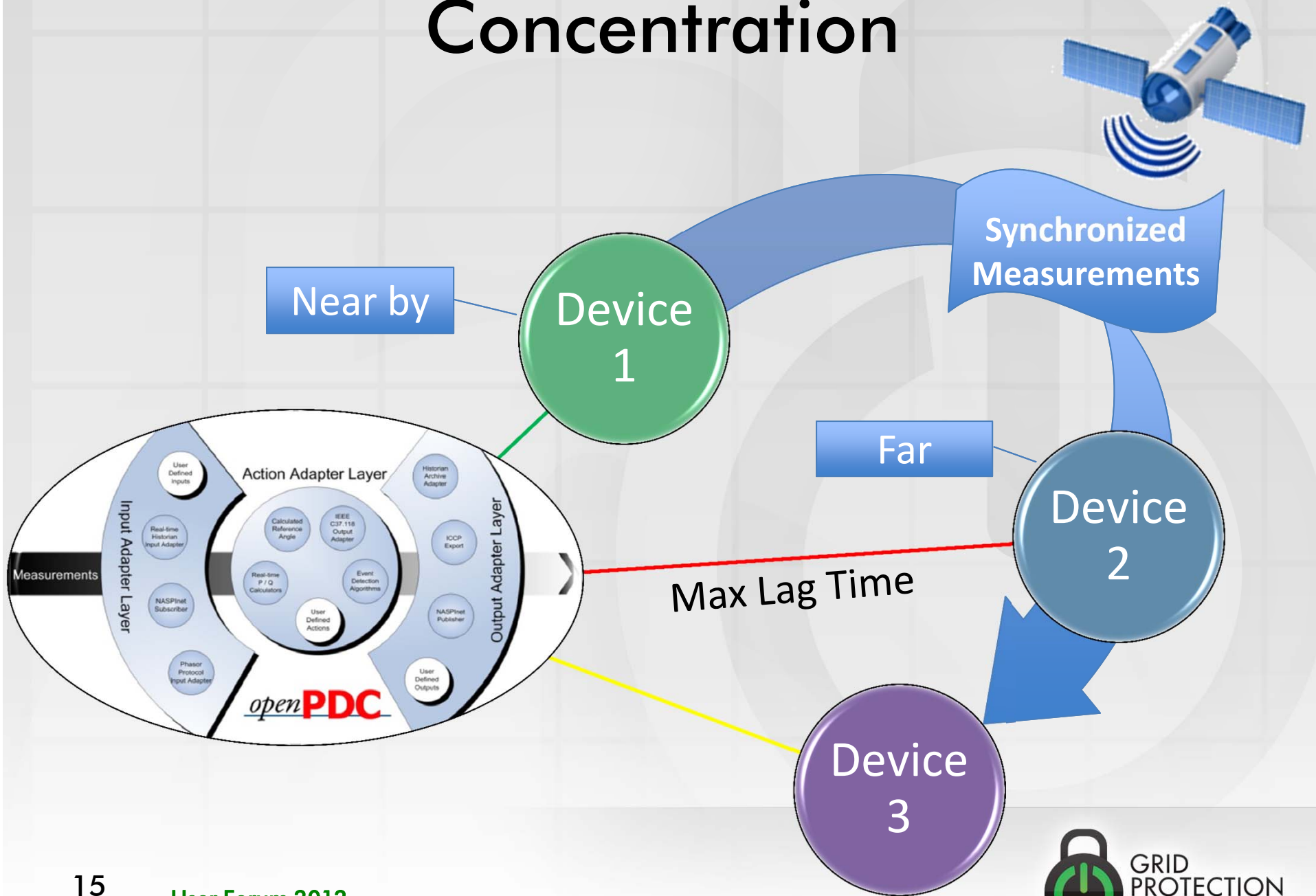
- Queue up measurement data for transmission to archival systems.

# Action Adapters

*Purpose:*  
**SORT**

- Sort measurement data by time and process the data for same time-slice.

# Concentration



# Data Flow Exercise

