# Project Partners

## Advanced Synchrophasor Protocol Project

**sttp**

**Streaming Telemetry Transport Protocol**

### Project Collaborators

<table>
<thead>
<tr>
<th>Project Collaborators</th>
<th>Project Financial Partner</th>
<th>Vendor</th>
<th>Utility</th>
<th>Demonstration Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonneville Power Administration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridge Energy Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dominion Energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electric Power Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electric Power Research Institute</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERCOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grid Protection Alliance (Prime)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISO New England</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MehtaTech</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oklahoma Gas &amp; Electric</td>
<td></td>
<td></td>
<td></td>
<td>WSU</td>
</tr>
<tr>
<td>OSIsoft</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak Reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PingThings</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PJM Interconnection</td>
<td></td>
<td></td>
<td></td>
<td>EPG</td>
</tr>
<tr>
<td>Southern California Edison</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>San Diego Gas &amp; Electric</td>
<td></td>
<td></td>
<td></td>
<td>WSU</td>
</tr>
<tr>
<td>Schweitzer Engineering Laboratories</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Southern Company Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Southwest Power Pool</td>
<td></td>
<td></td>
<td></td>
<td>WSU</td>
</tr>
<tr>
<td>Space-Time Insight</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trudnowski &amp; Donnelly Consulting Engineers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utilicast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tennessee Valley Authority</td>
<td></td>
<td></td>
<td></td>
<td>WSU</td>
</tr>
<tr>
<td>University of Southern California</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V&amp;R Energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Washington State University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Notes:**
- The table includes 26 project partners.
- There are 11 vendors.
- 11 utilities are listed.
- 6 demonstration hosts are identified.

---

*NASPI Work Group Meeting - Philadelphia, PA - October 23, 2018*
Deliverables

- Update PMP
- Release ASP Spec
- Develop α Toolkit
- Develop Demo Plan
- Publish Demo Results
- Publish API
What Makes STTP Different?

- Synchronized Frames vs Atomic Packets
- Reduced Data Loss
- Lossless Compression
- Scalability (to hardware limits)
- Publish / Subscribe Model
- Publisher Data Access Control
- IP Level Security
- Configurable Connection Origin
### Difference: Synchronized Frames vs Atomic Packets

**IEEE C37.118 /**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNC</td>
<td>Synchronization Field</td>
</tr>
<tr>
<td>FRAMESIZE</td>
<td>Frame Size Field</td>
</tr>
<tr>
<td>IDCODE</td>
<td>Identification Code Field</td>
</tr>
<tr>
<td>SOC</td>
<td>Station Code Field</td>
</tr>
<tr>
<td>FRACSEC</td>
<td>Fractional Seconds Field</td>
</tr>
<tr>
<td>STAT</td>
<td>Status Field</td>
</tr>
<tr>
<td>PHASORS</td>
<td>Phasor Values Field</td>
</tr>
<tr>
<td>FREQ</td>
<td>Frequency Field</td>
</tr>
<tr>
<td>DFREQ</td>
<td>Derivative Frequency Field</td>
</tr>
<tr>
<td>ANALOG</td>
<td>Analog Values Field</td>
</tr>
<tr>
<td>DIGITAL</td>
<td>Digital Values Field</td>
</tr>
<tr>
<td>Repeat Field</td>
<td>Repeat Field of Measurements</td>
</tr>
<tr>
<td>CHK</td>
<td>Checksum Field</td>
</tr>
</tbody>
</table>

**Data Packet Command**
- **MTU** (Maximum Transmission Unit)
- **Fragment**

**STTP Commands**

**Data Packet Command payload is a set of Data Points:**

1. **Data Point**
   ```
   struct {
     uint32 id;
     uint8[] value;
     uint8[] state;
   } DataPoint;
   ```

2. **Data Point**
   ```
   struct {
     uint32 id;
     uint16 length;
     uint8[] payload;
   } Command;
   ```

3. **Data Point**
   ```
   struct {
     uint32 id;
     uint8[] value;
     uint8[] state;
   } DataPoint;
   ```

**Command Code = 0x06**

**Individual Measurements**
- Measurements in one packet may differ from another

**1 Packet, Several Unsynchronized Measurements**

**NASPI Work Group Meeting - Philadelphia, PA - October 23, 2018**
Difference: Reduced Data Loss

Transmitting Data with Small Atomic Packets and no Concentration Reduces Data Loss

Data Loss

- **UDP Large**: 2.12%
- **TCP Large**: 0.22%
- **UDP Medium**: 0.31%
- **TCP Medium**: 0.03%
- **UDP Small**: 0.12%

**IEEE C37.118**

Data from testing performed at PeakRC
IP based connections use TCP for commands and optionally UDP for data transmission:

- **TCP** provides reliable communications allowing for high-yield stateful compression.
- **UDP** can be used for data transmission with the potential for UDP data loss and with less compression than TCP.

*Methods to implement STTP in Unicast/Multicast only configurations will be documented for use cases where a “no command” based STTP may represent a preferred option over Unicast/Multicast IEEE C37.118.*
Difference: Scalability

- **IEEE C37.118 V1 & V2 configuration frame size max out (65K). A second stream must be created.**
- **Increased data loss and latency.** Purpose-built / allocated networks typically required.
- **Data loss and latency issues begin to appear.** Network tuning may be required on common hardware.
- **Issues, if any, are easy to resolve.**

STTP handles from 3 to 5 million points per second per connection.
Difference: Publish / Subscribe Model

- **A-Data (Priority 1)**
- **B-Data (Priority 2)**
- **Metadata**

Routing Processor:
- Only data intended for subscriber will be published.
- The STTP data is interleaved.

Publisher → ROUTING PROCESSOR → Subscriber

Individual Data Point
Difference: Publisher Data Access Control

STTP Publisher

STTP Subscriber A

STTP Subscriber B

Green or Blue Measurements

Green or Yellow Measurements

All Source Measurements

Measurement 1
Measurement 2
Measurement 3
Measurement 4
Measurement 5
Measurement 6
Security at IP Layer

- **TCP**: Primary security is added at the socket using industry standard Transport Layer Security (TLS or SSL). X.509 certificates are used to authenticate connections and provide encryption through public key infrastructure.

- **UDP** (optional): When existing command channel is secured with TLS, UDP uses AES symmetric encryption with keys exchanged over the TLS secure channel.
Difference: Configurable Connection Origin

- Two Types of Connections Supported
  - **Forward**
    - *Subscriber connects to Publisher* – typical operation where a listening server-based publisher with connecting client-based subscribers
  - **Reverse**
    - *Publisher connects to Subscriber* – operation where client-based publisher connects to listening server-based subscriber; used to cross security zones in desired direction

- Bidirectional Communications Allowed
  - Once connection is established, publisher/subscriber functions can operate in either direction over the single connection
Publisher and Subscriber operations are “functions” in STTP – not “objects”.

As such, a publisher “sends” data and a subscriber “receives” data – always.

Crossing Security Zone with Reverse Connection
How is the Project Team Advancing STTP?

- Documenting the Specification
- Developing the General Use API
- Conducting Demonstrations
- Participating in IEEE Standardization
Specification development is open on GitHub:

- https://github.com/sttp/Specification

Daily builds of specification are available in PDF, HTML and GitHub markdown formats

Topics include:

- Protocol Overview
- Establishing Connections
- Commands and Responses
- Compression
- Security
- among others

Anyone can propose an edit with a pull-request

- See “How to Contribute” on spec site for details
Advancement: Developing the General Use API

Key STTP Requirements:
- Performant Data Exchange at Scale
- Extensible Metadata
- Access Control and Security
- Bidirectional Connectivity
Advancement: Developing the General Use API

Publisher

- **Methods**
  - Connect
  - DefineMetadata
  - Disconnect
  - DisconnectSubscriber
  - SendData

- **Callbacks / Events**
  - SubscriberConnected
  - SubscriberSessionEstablished
  - SubscriberDisconnected

Subscriber

- **Methods**
  - Connect
  - Disconnect
  - RequestMetadataTables
  - RequestMetadata
  - Subscribe
  - Unsubscribe
  - SecureDataChannel

- **Callbacks / Events**
  - ReceivedMetadataTables
  - ReceivedMetadata
  - ReceivedDataPoints
Advancement: Developing the General Use API

- **Core DataPoint Metadata**
  - Point ID (guid)
  - Device ID (guid)
  - Tag (string)
  - AlternateTag (string)
  - Description (string)
  - Enabled (bool)
  - Created (date-time)
  - Updated (date-time)

- **Device Metadata**
  - Device ID (guid)
  - Name (string)
  - etc.

- **Synchrophasor Metadata**
  - Point ID (guid)
  - SignalReference (string)
  - Protocol (string)
  - SignalType (string)
  - EngineeringUnits (string)
  - PhasorType (string)
  - Phase (string)
  - DataRate (float)
  - etc.

---

**Low-Level API**

Example Metadata
Complex Structure Encoding (e.g., IEEE C37.118)
- Includes, as needed, data concentration at final consumer
Advanced Use Cases

- **Advanced Data Logic**
  - Variable distribution of redundantly measured values
  - Blue-sky state data reduction (for apps that desire this)

- **Gateway transmission of other protocol data**
  - ICCP, DNP3, Modbus, OPC, OpenFMB

- **Dynamic Data Volume**
  - Adjust data publication volume based on system conditions, e.g., sending more information when an event has been detected for increased monitoring and detail (where desired)
WSU Tools Demo
- Tools testing and observations with STTP, e.g., memory utilization, data loss, CPU loading and impacts on running analytics
- Testing at TVA, SPP, OG&E and SDG&E

EPG Tools Demo
- Integration and comparison testing of ePDC for observations with STTP, e.g., CPU loading, data loss, latency and memory impact when receiving STTP data
- Testing at PJM and Dominion

EIDSN Transfer Demo
- Testing TVA → SPP with
  - IEEE C37.118, and
  - STTP over TCP
- Recording and comparing results with a real-world data transfer
This year the IEEE P10 STTP working group was established to develop a project authorization request (PAR).

The PAR was approved by the IEEE-SA New Standards Committee on September 27, 2018 and given a proposed IEEE standard number of **P2664**.