- **Milestones:**

Y1Q1
- **M01**: Create prototype ERSAP configurations for INDRA and CLAS12 test systems
- **M02**: Identify or capture SRO formatted data from CLAS12 and INDRA test systems with data tag/filtering capability (output data ready for further offline processing)
- **M03**: Evaluate existing solutions for configuring and launching remote distributed processes
- **M04**: Establish code repository(s), project site, and method of documentation

Y1Q2
- **M05**: Create stream splitter program for EVIO or HIPO data formatted files
- **M06**: Create stream splitter program for simulated data in PODIO for ePIC
- **M07**: Create VTP emulator using files produced by stream splitter
- **M08**: Create controller program to synchronize multiple VTP emulators

Y1Q3
- **M09**: Determine appropriate schema for all aspects of monitoring system.
- **M10**: Establish databases for monitoring system using existing JLab servers.
- **M11**: Integrate Hydra as monitoring component.

Y1Q4
- **M12**: Integrate off-line data analysis framework into platform for CLAS12 data
- **M13**: Integrate off-line data analysis framework into platform for ePIC or GlueX simulated data
- **M14**: Integrate example JANA2 analysis into platform

(Year 2 Milestones revised for Y2 Proposal)

Y2Q1
- **M15**: Establish general framework for RTDP simulation
- **M16**: Create configurable CPU proxy component
- **M17**: Create configurable GPU proxy component (hardware and software)
- **M18**: Create configurable FPGA proxy component (hardware and software)

Y2Q2
- **M19**: Create functioning hardware GPU component (e.g. CLAS12 L3)
- **M20**: Create functioning hardware FPGA component (e.g ML4FPGA)
- **M21**: Configure simulation of full SRO system using existing JLab hardware resources

Y2Q3
- **M22**: Establish working test of system that transfers >=100Gbps from CH to compute center
- **M23**: Establish working test of system that includes GPU component for portion of stream
- **M24**: Establish working test of system that includes FPGA component for portion of stream
- **M25**: Test system with remote compute facility (e.g. BNL or NERSC) at limits of available resources

Y2Q4
- **M26**: Configure system that results in stream(s) being received by JLab from external source
- **M27**: Collaborate with HPDF group to evaluate processing SRO data at JLab for external experiments
- **M28**: Complete documentation for platform to be used by non-experts

Objective 1: **Process Launcher:** The software that will be used to configure and launch each component does not itself need to be highly performant. The highly performant and specialized codes will already be encapsulated in the components themselves that that will have been developed outside of this project (sans objectives 3 and 4 below). The most appropriate language for this will be something like python3 or possibly julia. The configurations themselves should be expressible in a static file format to allow other tools for editing and visualization of the configurations to be developed in the future. A configuration format such as YAML would be a good choice as it is widely supported across numerous programming languages. For inter-process communication that will need to support both local and wide area networks, ZeroMQ or a similarly common, open source product will be used.

Objective 2: **Monitoring System:** The monitoring system will be developed using an appropriate set of existing free or open source tools such as Prometheus, and Grafana. Both Prometheus and Grafana are already in use in other places at JLab which will allow leveraging local expertise, installations, and network configurations. Additional visualization tools will need to be developed as part of this project. These will be web-based were appropriate, though some specialized tools may also be needed. Advanced monitoring for continuous data validation will also be needed. Use of the AI/ML based Hydra [6] monitoring system would be appropriate for this purpose.

Objective 3: **Proxy Components:** The proxy components that will allow effects-based simulations will be developed using a performant language such as C++. These components will need to read in and write out data streams to mimic the operation of the real component that does not currently exist. For example, a proxy component that is used to represent a reconstruction algorithm that is expected to read data from a single stream, perform X Mflops/kB, and write roughly 1/2 of the data to the output. The proxy would need to understand the header information of the incoming stream enough to modify it for the output, but would not need to understand the payload. It would also need to exercise enough dummy operations on the CPU cores to mimic the X Mflops/kB it was configured for.

Objective 4: **Multi-stream Event Source:** The multi-stream event source will need to be written in a highly performant language such as C++ or possibly Java. Data will be read from a file that is either in an experimental raw data format such as EVIO or simulated event data format such as ROOT. For data that is not already in a format that includes DAQ system indexing (e.g. simulated data) it will need to apply an inverse translation table to convert from detector component indexing. The DAQ indexing is needed to identify the crate/slot/channel element the data would have originated from so it can be sent over the appropriate output stream so as to mimic live data. This component will require multiple processes spread over multiple compute nodes in close coordination in order to achieve the high bandwidths needed.

Objective 5: **High Bandwidth Test:** Configuring a full scale system that includes both real and proxy components and testing it at high bandwidth is necessary to demonstrate the platform's core functionality. Current expectations are to have a 400Gbps link available between the Hall A,B,C counting house and the Computer Center in CEBAF Center sometime in FY2024. The high speed testing will be coordinated to occur when the beam is down so that the full bandwidth will be available for the testing periods. The SoLID experiment serves as an example of the type of high bandwidth experiments being anticipated to run at JLab in the future. It will then serve as a useful guide for the testing configuration, even if the configuration is not an exact match for SoLID. There are currently eight U280 FPGA cards in the Computer Center purchased for use with the EJFAT project which would be available to use for these tests. Similarly, the Scientific Computing farm will have a few dozen GPUs (mostly Tesla T4's) available that could also be utilized for these tests. Utilizing real hardware components will be an important part of the platform and so will need to be included for the full scale configuration testing. We will utilize existing components developed outside of this project to exercise the heterogeneous components. For example, PHASM, CLAS12 tracking, and the EIC R&D project: ML4FPGA [12].

Objective 6: **Insights for HPDF:** The platform will be a tool for developing, testing, and validating SRO systems that utilize remote compute facilities. Here, "remote" can mean the Computer Center relative to the Counting House. Similarly, the HPDF is expected to serve as a remote compute facility for experiments outside of JLab. The proposed platform tool will provide valuable insights into how to best use the HPDF to support experiments. The approach to obtaining this objective will be to partner with a remote facility such as BNL or NERSC to perform some limited testing once the project matures. Exercising a high bandwidth application where the data originates at JLab and is sent to a remote compute facility will give valuable insight on how the HPDF might handle data streaming in from a remote site. A reach goal will be "bouncing" the data stream(s) we send to a remote site back to the HPDF as an additional testing phase.