**PROGRAM PROPOSAL**

**Proposed Program Name:** Plexus Interop - <https://symphonyoss.github.io/plexus-interop/>
**Sponsoring Member(s):** Deutsche Bank, Thomson Reuters (Saori Fotenos)
**Initial PMC Lead:** Rhyddian Olds, Deutsche Bank
**Proposed Scope:** The Plexus Interop project aims to define an open standard for desktop application interoperability and provide a container-agnostic reference implementation of this standard. It formalizes connections between applications within a single user session on the client desktop.

The main goal is to enable development of extensible workflows connecting independent apps developed by different organizations in different technologies (.NET, Web, Java, Python, etc.) by passing relevant context (structured data objects) between those apps either by defining or utlising existing open standards.

The desktop application interoperability standards currently proposed by the programme are documented here: <https://symphonyoss.github.io/plexus-interop/#reference> and will be maintained by a Desktop Interop Working Group

Like many other Remote Procedure Call (RPC) systems, Plexus Interop is based around the idea of standards which allow a service to be defined by specifying the methods that can be called remotely with their parameters and return types. The Plexus Interop reference implementation mirrors the RPC methods supported by [gRPC](http://www.grpc.io/docs/guides/concepts.html#rpc-life-cycle) by implementing four interop standards:

1. Unary call – the consumer sends a request, provider receives, handles, and replies it.
2. Consumer streaming call – the consumer sends many requests, the provider sends one response.
3. Provider streaming call – the consumer sends one request, the provider sends many responses.
4. Bidirectional streaming call - consumers send many requests, providers send many responses.

However, the standards go beyond the scope of existing RPC standards:

1. Application lifecycle management – standards for tracking the lifecycle of desktop apps, e.g. whether an app is running or not, how an app can be launched on demand.
2. App Launchers – standards for defining customisable installation, startup and shutdown flows.
3. App Hierarchy – standards that will support interop between multi-app products and single-app products
4. Interop Registry- standards to specify which interop capabilities the app consumes and provides, allowing the tracking of API use critical to their evolution and management.
5. Central Broker - Although the industry is moving towards containerized desktop applications, the standards recognize that much of the application landscape is not yet tied to containers and is architecturally agnostic of container implementation. Separating interoperability from the container provides notable advantages: different containers can be leveraged in the same workflow, and launched applications residing outside of containers can participate in Interop activities.
6. Security – standards on how to secure interop between applications.

Initially the program focus will be on a working group to continue the development and maintenance of the reference implementation of these proposed standards (the Plexus Interop Project), and a working group to formalise and validate example Business use cases to measure the success of both the standards definitions and project implementations (the Interop Use Case Validation Working Group).

The sponsors encourage additional working groups and contributions that expand scope of the initial standards by the community. In accordance with the standard Program Operations Policy, the PMC will determine the creation and scope of additional working groups. Working groups will collaborate but operate independently of each other publishing relevant standards and optionally reference implementations. PMC will ensure to consider the full breadth of programs and working groups across industry standards to ensure we minimize overlapping standards.

Examples suggested so far for which the standards and reference implementation design allow are:

* Multi-machine – between apps run by the same user on different machines.
* Multi-broker – federating multiple interop brokers running on the same machine.
* Decentralised Interop Registry – federating interop registries

**Program Operations Policy: Standard:** ☒
 **Amended:** ☐ (**please provide redlined POP with this proposal)**

**Initial Projects and/or Working Groups:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Scope** | **Initial Roster** |
| Desktop Interop Working Group | Maintain the open standards for desktop application interoperability documented here: <https://symphonyoss.github.io/plexus-interop/#reference> | Slava Kryukov, Deutsche Bank (chair)Jorge Santos, Thomson Reuters |
| Plexus Interop Project | Provide a container-agnostic reference implementation of the Desktop Interop standards. | Anton Nikolaev, contributorMikhail Udalov, contributor |
| Interop Use Case Validation Working Group | Define and publicize specific interoperability scenarios required by participants to be used as target success criteria in the implementation of the Interop standards. | Saori Fotenos, Thomson Reuters (chair)Slava Kryukov, Deutsche Bank |