SYMPHONY

MARCH 2022

Connecting Symphony with Desktop Integration Platforms

Technical solutions that integrate Symphony with Desktop Integration Platforms

Bertrand Laporte, Lead Front-End Architect

Dimiter Georgiev, Product Manager

Pierre Neu, Product Manager

What is a Desktop Integration Platform?

> Symphony considers DIP integration as a key workflow enabler and is actively working with the FINOS FDC3 working group to standardize DIP integration APIs across the industry.

Desktop Integration Platforms (DIPs) are client-side integration solutions that facilitate more efficient workflows for financial professionals who navigate between many desktop applications on a daily basis.

From a functional point of view, DIP integrations provide many benefits, such as:

- Cross-application workflows (e.g. initiate a chat conversation from a CRM);
- The ability to minimize cross-application copy/paste by sharing context (structured or unstructured) or sending "intents" (e.g. share a formatted message in an existing chat);
- Better management of screen real-estate (DIPs can arrange multiple application windows into workspaces); and
- Aggregated notification centers with support for rich and actionable notifications.

As explained in a previous **publication**, Symphony considers DIP integration as a key workflow enabler and is actively working with the FINOS **FDC3** working group to standardize DIP integration APIs across the industry.

Technical Challenges

Even though DIP providers aim to offer flexibility to connect to their platforms, a few technical challenges need to be addressed to integrate an application like Symphony.

Transparent Integration

The first challenge is that Symphony is not a customer-specific solution, but a universal platform designed to work under different configurations. Consequently, Symphony must support all DIP providers. In addition, the Symphony DIP integration has to be as lightweight and flexible as possible. In other words, it should be possible to enable the integration through a simple configuration change and should not require complex software updates or dedicated software versions. Finally, we want to keep our client software modular and avoid unnecessary code for customers that don't use a DIP integration - i.e. the Symphony desktop client should not even download resources dedicated to DIP integration if such integration is not activated.

This is why Symphony has selected a technical solution based on optional **sandboxed plugins** transparently deployed upon configuration changes. These plugins will be developed and validated by Symphony. They will be hosted with the Symphony code base so their deployment will not require any IP whitelisting or network configuration changes.

> Symphony has selected a technical solution based on optional sandboxed plugins transparently deployed upon configuration changes.

Security

As security is one of Symphony's core tenets, our approach to DIP integration must not introduce any risks in this area. DIP providers usually offer two integration mechanisms for applications based on web technologies like the Symphony client:

- The first mechanism, called *embedded mode*, requires the DIP provider to launch the web application as a web view in one of its window containers.
- The second mechanism, called *standalone mode*, allows the DIP provider to work with applications launched as standalone processes and interact through inter-process communication. This mechanism involves integrating proxy libraries (provided by the DIP) to communicate with the DIP agent through DIP-specific and FDC3 APIs.

From a technical standpoint, the embedded mode is much simpler to implement and is probably the best option for applications that do not manage sensitive information. However, web applications running in DIP webviews do not run in the browser's sandboxed environment, which means that code running in the DIP can technically access all data loaded in the webapp memory. Importantly, this would compromise Symphony's security chain, such as messages and chat cryptographic keys, which is why embedded mode is not the default Symphony integration mechanism.

As previously mentioned, our integration solution is to rely on optional plugins (also known as adapters) that will allow communication with any DIP, while still running the Symphony client as a standalone process. A secondary benefit of this approach is that running as a standalone application also ensures better performance: the Symphony client can run in the background, enabling it to quickly react to any client-side events.

> Security is one of Symphony's core tenets, our approach to DIP integration must not introduce any risks in this area.

Symphony DIP Adapters

Overview

The DIP adapters are optional plugins that will extend the Symphony client capabilities and will be dynamically downloaded through activation switches in the Symphony Admin and Compliance Portal.

The goal of these plugins will be to open communication channels between the Symphony client and the DIP APIs.

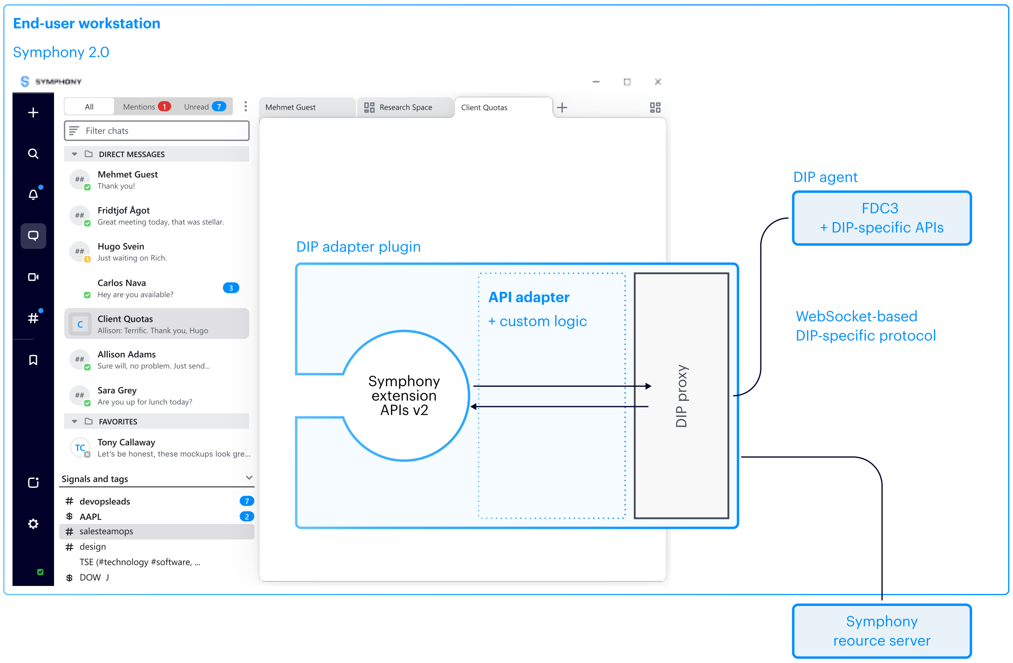
Technically, they will be loaded as headless sandboxed iframes and downloaded from the Symphony resource servers along with the rest of the Symphony client code. As such, no specific changes will be required in the customer environment as the plugin code will be downloaded like any other Symphony resources. In addition, the adapter code will be managed by Symphony in partnership with the DIP providers.

> The goal of these plugins will be to open communication channels between the Symphony client and the DIP APIs.

From a communication standpoint, the plugins will provide a translation layer between the Symphony and the DIP APIs. As such they will have two connection points:

- First, they will connect to an improved version of the Symphony client APIs that expose events and services required to send or react to FDC3 events. For instance, this client API will allow the adapter to submit formatted messages for a given conversation. It will also notify the adapter when certain events (like notifications or certain user actions) occur in the Symphony client application.
- Second, they will connect to the DIP agents through proxy libraries provided by the DIP providers. Behind the scenes, those proxy libraries will communicate with the DIP agents through secured WebSocket connections (using DIP-specific protocols).

GENERAL OVERVIEW OF THE ADAPTER PLUGIN



Mehmet Guest
Thank you!

Fridtjof Ågot
Great meeting today, that was stellar.

Hugo Svein
Just waiting on Rich.

Carlos Nava 3
Hey are you available?

Client Quotas
Allison: Terrific. Thank you, Hugo

Allison Adams
Sure will, no problem. Just send...

Sara Grey
Are you up for lunch today?

Tony Callaway
Let's be honest, these mockups look gre...

devopsleads 7
$ AAPL 2
# salesteamops
# design
TSE (#technology #software, ...
$ DOW J

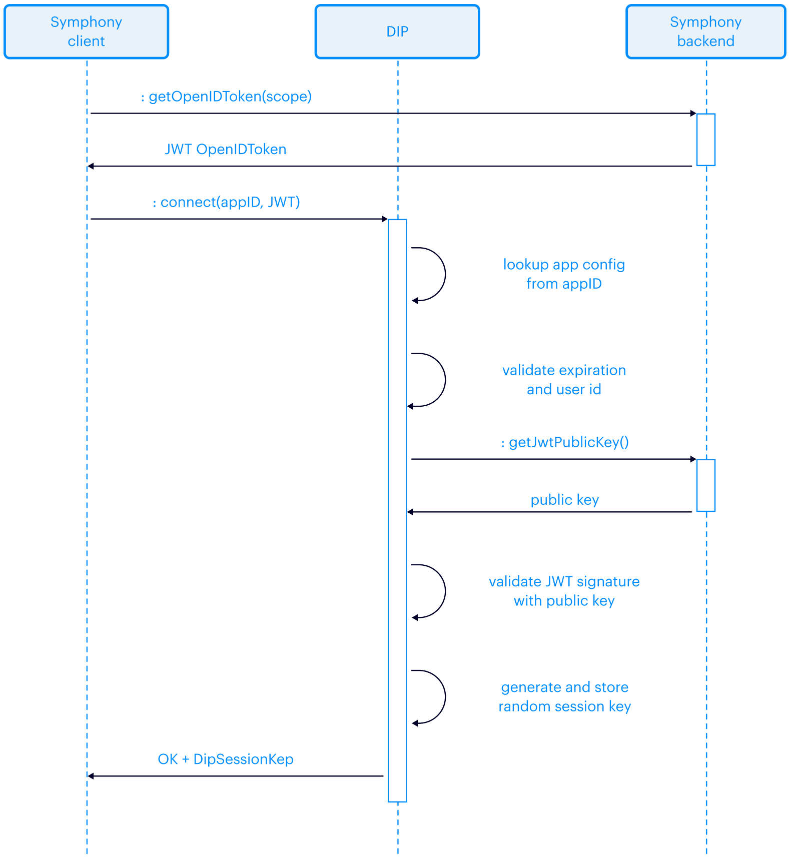
The plugin logic will be simply to:

- Listen to Symphony events and transform them into DIP API calls, such as FDC3 intents, FDC3 context sharing, or DIP proprietary API calls; and
- Listen to DIP events and transform them into Symphony API calls – for instance, to create a new chat, bring a chat room into focus, or send new chat messages containing FDC3 context data (e.g. ticker).

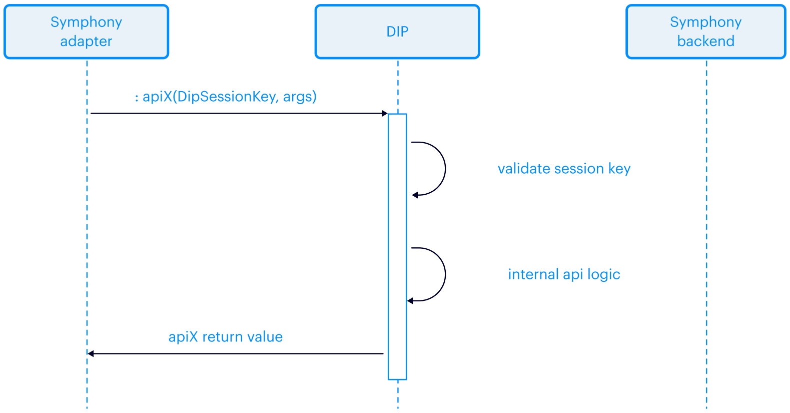
To make this possible, the Symphony client will expose a new TypeScript-compatible API that will come on top of the current client-side extension APIs. This new API will improve the developer experience. It will expose new services and events – like the capacity to submit messages, which is not offered today – and will also integrate a finer permissions management system to select which events or data can be shared with the DIP's agent.

This new client API will also become available to future client-side extensions (also known as Symphony extension apps).

DIP ADAPTER PLUGIN



API CALL WITH SESSION KEY VALIDATION



Authentication

From the DIP perspective, the Symphony application is a new client that must provide proper authentication to connect to the DIP agent.

As Symphony's objective is to support all DIP providers through a common set of APIs, we will move to an **OpenID** authentication model which is standard and provider-agnostic.

During the connection phase, the **OpenID token** will be passed to the DIP agent to authorize the secure WebSocket connection. The DIP agent will have to authorize the connection by validating the token expiration date as well as its claims and digital signature (this will be done with the Symphony public signing key that will be either provided to the DIP configuration or exposed through a public Symphony Back End service accessible to the DIP).

Once the connection is authorized, the DIP will have to generate a random session key that will be passed to any subsequent calls to the DIP internal APIs

First FDC3 Integration Use Cases

The first FDC3 integrations supported by the Symphony client will cover the following topics.

As input (i.e. messages received from the DIP agent):

- Creating new chat rooms via the **StartChat** intent
- Sending a message into existing chat rooms via the **SendChatMessage** intent

In both cases, the Symphony client will display a confirmation dialog to let the end user fine-tune the chat creation (i.e. room name, member lists) and/or the chat messages.

As output (i.e. messages sent to the DIP agent):

- Submitting new intents like **ViewInstrument**
- Sharing context data (e.g. an **instrument**) in an FDC3 channel

Conclusion

Symphony is deeply committed to developing and improving its client-side DIP integration capabilities.

As explained, DIP adapters will come as native Symphony plugins that will be dynamically installed or removed through simple configuration changes. Adapter plugins will allow leveraging of all DIP capabilities, as there will be one adapter per DIP provider, which means both standard and proprietary APIs will be supported. This modular approach allows Symphony customers to choose any DIP provider to support standard FDC3 integration.

Our first beta version will be available in the first half of 2022 and is designed to be compatible with all current DIP providers.

SYMPHONY

Neither Symphony Communication Services, LLC (together with its affiliates, "Symphony") nor the services which Symphony offers to its customers ("Symphony Services") is subject to the oversight or jurisdiction of any financial regulatory body.

Symphony is not, and shall not be construed as, a registered broker-dealer, an underwriter of securities, an introducing broker, or an entity performing similar or other regulated functions. Similarly, Symphony is not registered as, or to be construed as, a multilateral facility (including but not limited to a SEF, MTF or OTF), trading venue, regulated market or exchange. Because Symphony does not act in these capacities, nor is Symphony registered to do so, any indication to the contrary in this document or in any other document provided to you by Symphony shall be interpreted to the maximum extent so as not to imply that Symphony is acting in such capacity (or in any other capacity that is subject to regulatory oversight or subject to a requirement to register with a financial regulator).

Any Symphony Service that permits customers to negotiate the terms of any financial instrument is offered, and only capable of being used, on a purely bilateral basis. Symphony has no role or involvement in the negotiation or arranging of any transactions in financial instruments nor in the introduction of counterparties to one another for purposes of entering into such transactions. The execution of any such transaction is solely as between the counterparties thereto, and Symphony does not perform any execution, clearing or settlement function for any financial instrument transaction. To the extent that any Symphony customer utilizes Symphony Services for purposes of negotiating financial instruments, it is solely the responsibility of such customer to comply with any regulatory requirements to which such customer, or such negotiation, may be subject.

Neither this document, nor the offering by Symphony, or use by any Symphony customer, of any Symphony Service, shall be construed as a solicitation, endorsement, offer or recommendation by Symphony to buy, sell or hold any financial instrument, or an offer to provide any investment advice or investment services to any person in any jurisdiction. Recipients of this document and users of all Symphony Services alone assume the responsibility of evaluating the merits and the risks associated with the use of the Symphony Services.

Furthermore, certain of the statements and terms contained in this document may be forward-looking and contain concepts, product ideas or functionalities that are not currently part of the Symphony Service suite, but are instead contemplated by Symphony to be offered to customers at a future date. There is no guarantee that any such functionality will be offered by Symphony, and until such time, if any, as a particular functionality is included in the Symphony Services pursuant to a contractual agreement between Symphony and the customer utilizing such functionality, the depiction of such concepts, ideas and functionalities shall be construed as forward-looking only.

Nothing in this document or in the license or operation of any Symphony Service constitutes a representation by Symphony that is contrary to the foregoing disclaimers.

Learn more at www.symphony.com