



FDC3

Desktop Agent Bridging FINOS Members Meeting 2023



Kris West



Matt Jamieson



Aaron Haines



Tiago Pina

FDC3 and the Modern FS Desktop

FDC3 has helped evolve the Financial Services Desktop:

- Strong uptake at Banks, Buy-side firms & Large vendors
 - Suites of interoperating apps replacing monolithic applications
- Smaller vendors / dev teams publishing and integrating apps into workflows
 - Reduced hurdles / multilateral agreements needed for participation
- FINOS Conformance Testing program
 - Greater consistency → reduced vendor lock-in

Why?

- It's easier to agree on working in an open standard
- Less throwaway customisation
- Cross platform / vendor / technology support



But...

There are still some pretty big challenges reducing uptake!

What is it NOT achieving?

Many FDC3 adopters have an internally-assembled desktop suite, but they have other apps they want to interoperate with

Independent suites of vendor apps, delivered in their own container

- Integrate everything in your container?
- UX compromises?

Monolithic (usually non-web) apps

- Interop-only often preferred
- Desktop agent-specific .NET/Java interop APIs
- Stalling the integration of native apps

What is it NOT achieving? (2)

Application vendors need a lower barrier to integration

- Many existing financial services software vendors are keen to support FDC3
 - Their customers are equally keen!
 - Vendors are often the most impacted by the lack of a solution for large native apps:
 - Their apps are often sets of components,
 - already integrated into a container (whether from a vendor or a proprietary one), and
 - frequently built using .NET, Java or other non-web technologies
 - The cost of implementing and maintaining integrations to each desktop agent vendor, or customer's proprietary desktop suite is too high
 - A wire-protocol specification of FDC3 would significantly lower this cost

What is it NOT achieving? (3)

Users have many devices: multiple PCs, phones, tablets

- At many banks and buy-side firms, users already have multiple PCs
 - Their workflows span across desktops
 - The cross-desktop workflows are often the most inefficient and prone to error
- Users are also increasingly working remotely and from mobile devices
 - To achieve the goal of FDC3 we will eventually need to cater for mobile



FDC3

Desktop Agent Bridging

Proposed addition to the
FDC3 Standard

How are firms facing these challenges?

FDC3 doesn't currently have a good answer to any of this:

- Individual Desktop Agents do...
- But getting 3+ stakeholders to agree on one is a multilateral agreement...

The lack of a Standard solution makes the situation more complex:

- Multiple solutions
- Incomplete solutions (e.g. channels only)
- Islands of Interop
- Multilateral agreements break down hard-won value in FDC3

A vendor's & adopter's choice of Desktop Agent vendor should NOT limit their ability to interop!

Mar 5th 2019: FDC3 1.0

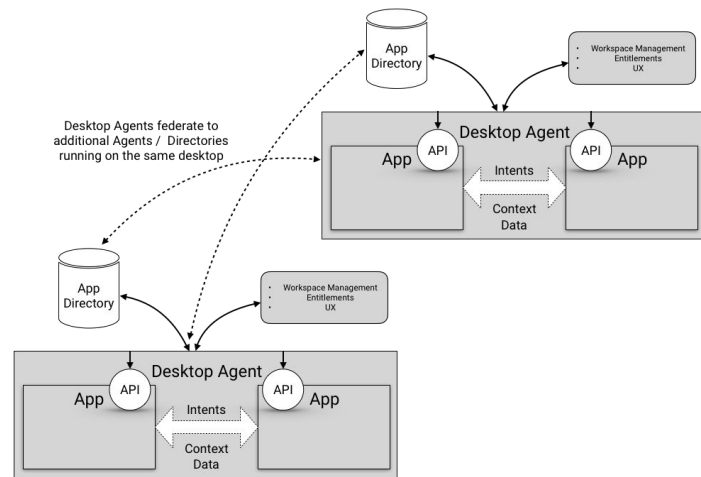
Inter-Agent Communication

A goal of FDC3 standards is that applications running in different Desktop Agent contexts on the same desktop would be able to interoperate. ...

... an App in one Desktop Agent context would not need to know a different syntax to call an App in another Desktop Agent context...

The actual connection protocol between Desktop Agents is not currently in scope for FDC3 standards. Given that there are a relatively small number of Desktop Agents, and that any given desktop will have a finite and relatively static number of Desktop Agents installed at any given time, the connectivity between different Agents can be adequately handled *for the time being* on a case-by-case basis.

<https://fdc3.finos.org/docs/1.0/api/api-spec#inter-agent-communication>



2.5 years later...

FDC3 Standards Working Group:

- **Sep 23rd 2021 #459:** Support the production of Desktop Agent Bridging #453

FDC3 Use Cases Roundtables

- **London, Oct 5th 2021:** Use cases that FDC3 should address: Cross-container communication of context
- **New York, Nov 9th 2021:** Container Fragmentation is holding up Buy/Sell side integration, need for many-to-many (containers) integration

FDC3 Standards Working Group:

- **Oct 28th 2021 #481:** Bridging is within FDC3's charter & work should be undertaken to achieve it
- **Nov 18th 2021 #505:** Use cases examined, existing solution demos, formation of a discussion group to develop proposals
- **May/June 2022 #717:** FDC3 2.0 adds recommended User channels to aid future bridging use

Desktop Agent Bridging Discussion Group

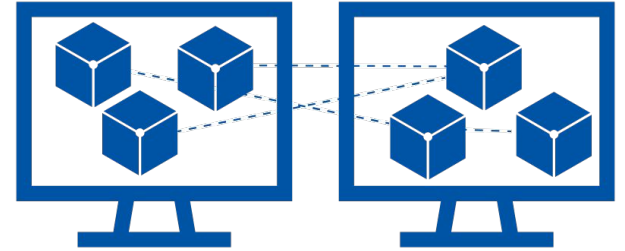
- 16 Meetings (Dec '21 - May '23)
- Participating firms:
 - NatWest
 - Citi
 - Cosaic/Finsemble/Interop.io
 - Glue42/Interop.io
 - OpenFin
 - Wellington
 - UBS
 - Factset
 - Whitedog
 - Symphony
 - RBC
 - FINOS
 - LSEG
 - Scott Logic
 - Expero
 - Kolbito
 - Singletrack
 - S&P Global
 - State Street
 - Adaptive

544 Desktop Agent Bridging #968

The screenshot shows the FDC3 website with a blue header. The navigation bar includes links for 'Getting Started', 'Roadmap', 'Community', 'The Standard', 'Get Involved', 'FINOS AppD...', and 'Training'. The main content area features the FDC3 logo and the tagline 'Open standards for the financial desktop'. Below this are five feature cards, each with an icon and a brief description. A yellow arrow points from the text '544 Desktop Agent Bridging #968' to the 'Agent Bridging' card.

API	Intents	Context Data	App Directory	Agent Bridging
Create a consistent developer experience by adhering to the API standard	Use standardized verbs to instruct other apps to take an action	Share context between apps to eliminate re-keying and streamline workflow	Discover trusted apps that can take part in a FDC3 workflow using an App directory	Link 2 or more Desktop Agent APIs together via a 'bridge' to extend interoperability across them. <i>@experimental</i>

Bridging API Overview



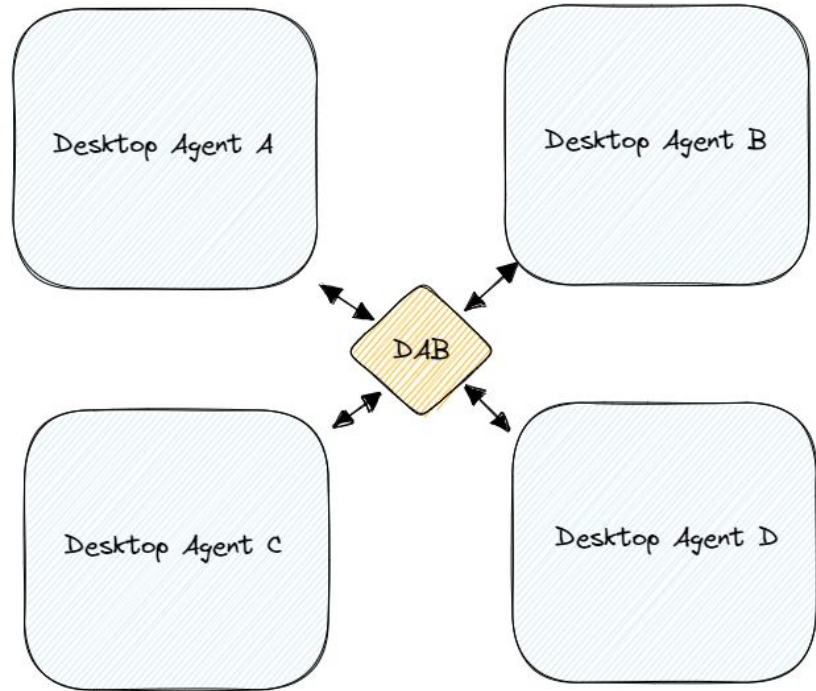
Bridging:

- Is a **JSON-based wire protocol** for **Desktop Agents** (rather than apps) to interoperate
 - Insulates apps / FDC3 API from additional complexity
- Is intended to connect DAs (and the apps they manage) **for a single user**
 - It does not connect users to each other!
- Can **connect different devices** (for that same user)
- Aims to **keep as much of the additional complexity in the bridge** as possible.

Bridging API Overview - Topology

The 'Bridge':

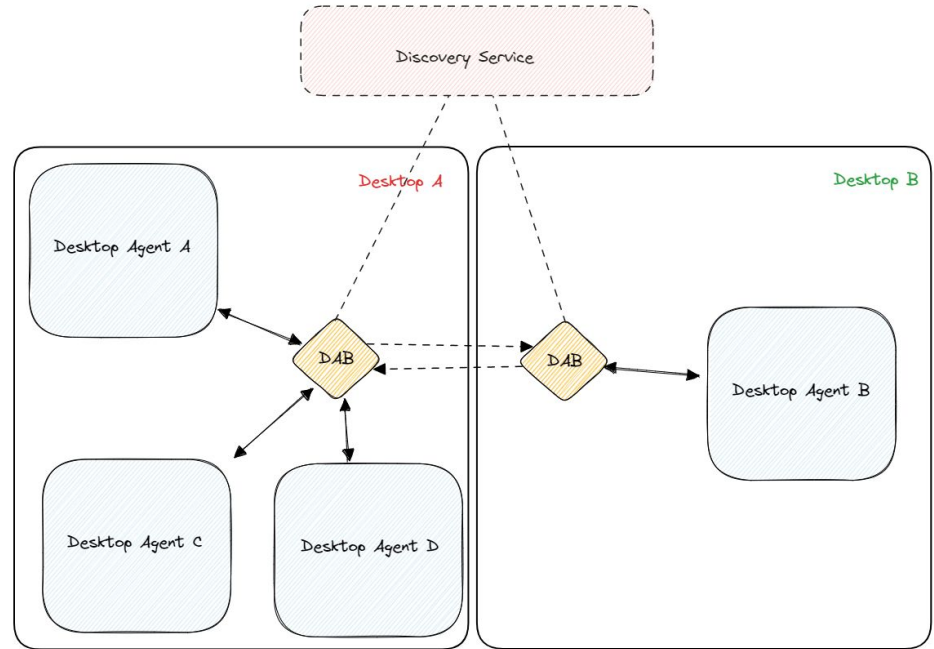
- Is its own entity (i.e. it's not part of a Desktop agent).
- Is owned / deployed / configured by the end-user firm.
- Acts as a 'server' which each Desktop Agent is a 'client' of.



Bridging API Overview - Multi-device

Multi-device Bridging is supported:

- It is the bridge's responsibility to handle single-user multi-machine cases
- For example, by offering plugin support to allow for customisation and additional functionality - e.g. a Discovery Service that works with internal IT infrastructure



Bridging API Overview - Components

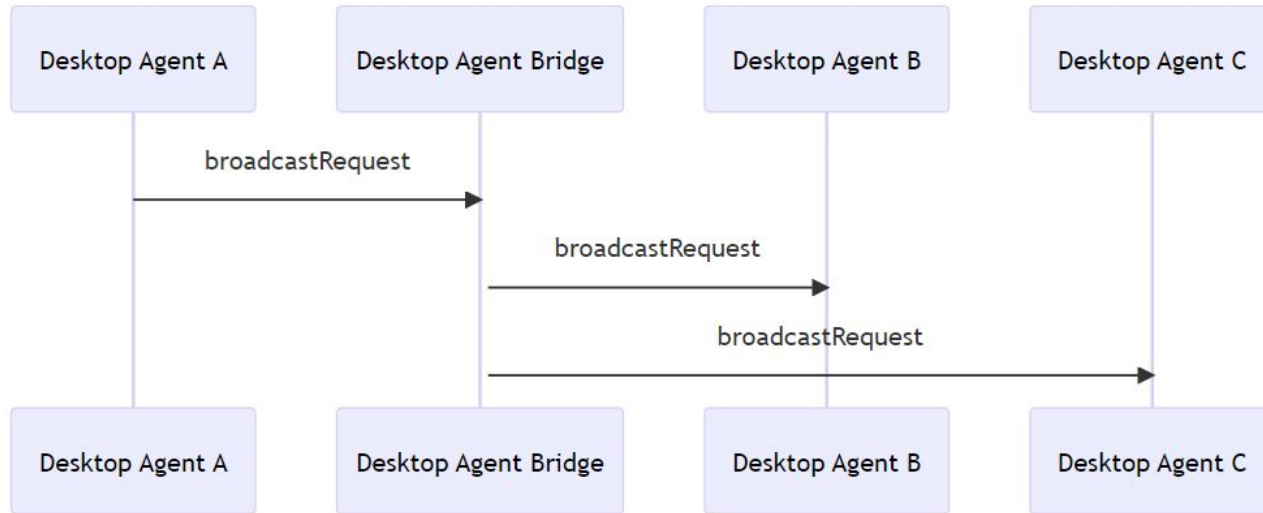
The Bridging specification is composed of 3 key components:

- **The Connection:** The pipe through which Desktop Agents communicate with the bridge:
 - WebSocket - perhaps other protocols in future (e.g. postMessage in a browser)
- **Connection Protocol:** Defines how a Desktop Agent connects to the Bridge
 - **connect** → **hello** → **handshake/auth** → **connectedAgentsUpdate** → **disconnect**
 - Includes:
 - Authentication
 - Name assignment
 - DA channel state synchronization
- **Messaging Protocol:** Defines message exchanges (encoded in JSON, and defined by in JSON Schema) to support each FDC3 API call

Bridging API Overview - Messaging Protocol

The Bridge acts as a router and message aggregator for several different types of message exchange:

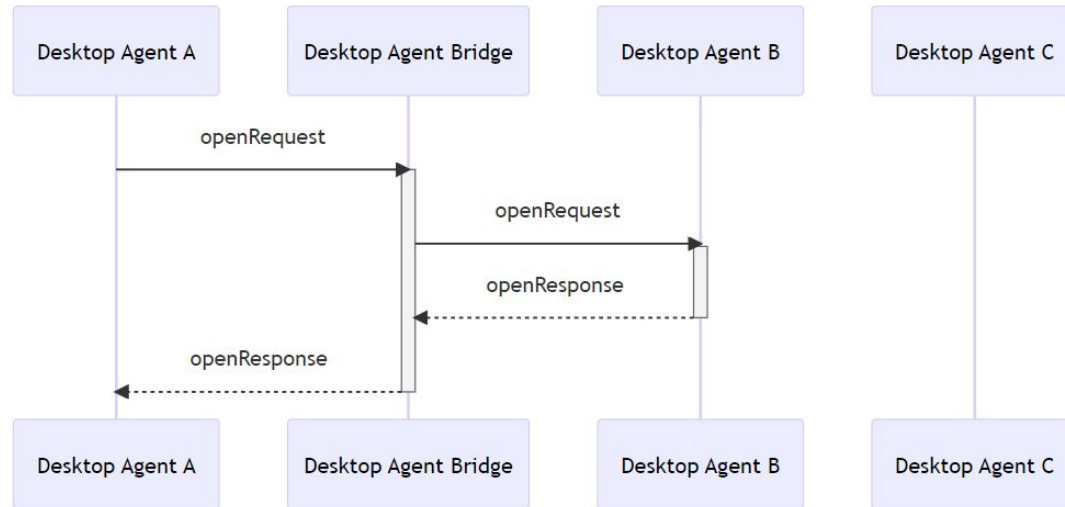
Request only: One-way communication, e.g. `fdc3.broadcast`



Bridging API Overview - Messaging Protocol

The Bridge acts as a router and message aggregator for several different types of message exchange:

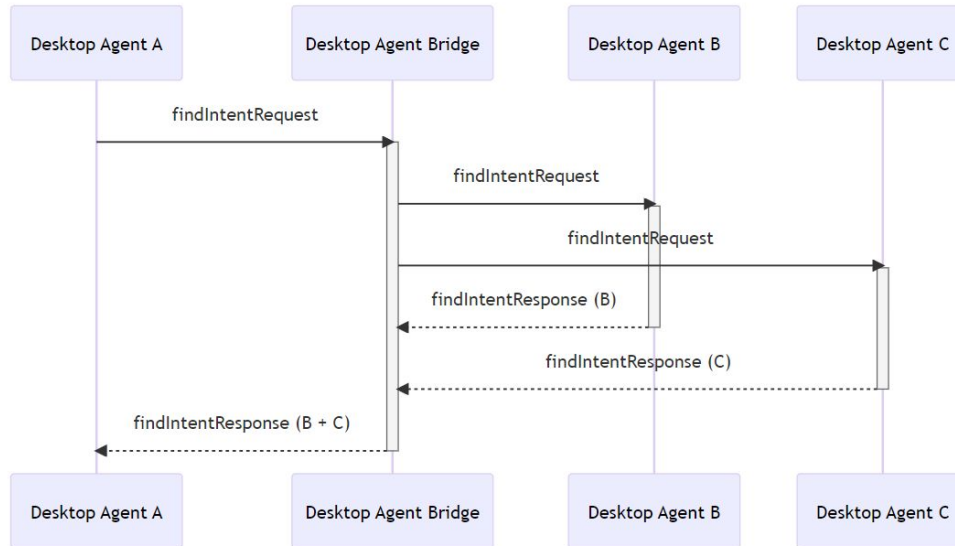
Request Response (single): Two-way comms with a single agent, e.g. `fdc3.open`



Bridging API Overview - Messaging Protocol

The Bridge acts as a router and message aggregator for several different types of message exchange:

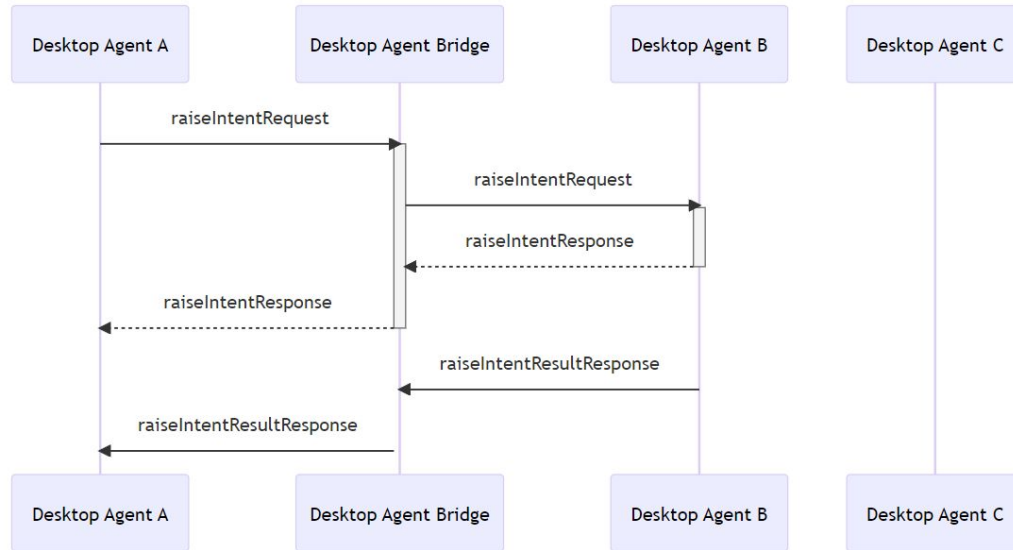
Request Response (collated): Collated responses from many agents, e.g. `fdc3.findIntent`



Bridging API Overview - Messaging Protocol

The Bridge acts as a router and message aggregator for several different types of message exchange:

Request Multiple Response (single): Multiple responses from one agent, e.g. `fdc3.raiseIntent`



DEMO

Courtesy of NatWest Agile Markets
& State Street GlobalLink



NatWest

STATE STREET.



Confidential

A Real World Problem

NatWest has a complex estate

- Apps in a range of technologies running inside and outside container & DA contexts - .NET, Java, Browser, OpenFin, Finsemble
- Workflows that span contexts and also span multiple machines

Examples

- Internal RFQ workflow - shares instrument context across multiple apps
- Sharing contact information from chat / telephony to display sales MI
- Customers want to link specific capabilities from Agile Markets into their workflows

Direct integrations don't scale - a standard helps us decouple and independently evolve

A Real World Problem

The screenshot displays a financial trading application interface with two main panels. The left panel, titled "Pricing Grid", shows a search for "LIBOR-Long" instruments, with "usswap" selected. A table lists instrument details for "GB00B128DH60" in GBP currency, with a desk of "infbond". A yellow tooltip with a red border and a close icon is visible over the search area, containing the text "Remove 'Jump To' filter: GB00B128DH60".

The right panel, titled "TradeBookings", shows a list of trade entries with columns for Status, Broker, Venue, and Side. A modal window is open for a trade with the following details:

- Trade: Buy - 0.0003M GBP - UKTI 1.25 11/22/27 3MO@1000 - null
- Trade Type: Buy
- Instrument: UKTI 1.25 11/22/27 3MO
- Amount: 0.0003M GBP
- Book: GARTR
- Ex. Time: 13-06-2023 10:58
- Ex. Trader: [Redacted]
- Jump to Pricing (highlighted)
- Sales Code: Z43
- Amount: 0.01
- Salesperson: [Redacted]
- Quality: TO

The modal also includes a "Trade Detail" section with fields for RBS Trader, AQ Trade Time, and Exec ID, and a "Fixed" section with dates and terms. A "Manual" button is visible at the bottom right of the modal.

Backplane

- Developed at NatWest as an FDC3 bridging solution
- Now Open Source and contributed to FINOS
- Work ongoing to align with the standard
 - contributions welcome!



Components

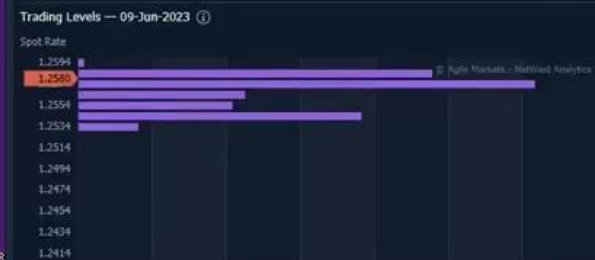
- Bridge Desktop Service
- JS Client Lib
- .NET Client Lib

<https://backplane.finos.org/>
<https://github.com/finos/backplane>

agile markets

Apps Shortcuts Notifications Aaron Haines

Blotter Currency Quote ECI reconciliation



FX Connect v7.25.1001

my.uat.globallink.com/fxconnect/orders/assigned

FX Connect v7.25.1...

FX Connect®

My FXC Orders 0

Outright \$0.0
Swap \$0.0

\$0.0 Gross USD

0 Unassigned
203 Assigned

0 All Sessions 0 Staged 0 Algo 0 RFS 0 RFS Auto 0 Allocation 0 Competitive

Assigned Orders | 26.2M Showing 203 of 203 (0 selected)

Assign to Me Unassign Assign to Other

<input type="checkbox"/>	Group	Platform	Order Type	Quote Pair	Side	Given CCY	Give
<input type="checkbox"/>		FXC	Outright	EUR/USD	BUY	EUR	39,6
<input type="checkbox"/>		FXC	Outright	USD/CAD	BUY	CAD	11,0
<input type="checkbox"/>		FXC	Outright	GBP/USD	SELL	GBP	20,9
<input type="checkbox"/>		FXC	Outright	GBP/USD	BUY	GBP	97,4
<input type="checkbox"/>		FXC	Outright	GBP/USD	BUY	GBP	11,6
<input type="checkbox"/>		FXC	Outright	USD/CAD	SELL	CAD	56,5
<input type="checkbox"/>		FXC	Outright	GBP/USD	BUY	GBP	35,5
<input type="checkbox"/>		FXC	Outright	USD/PHP	SELL	PHP	5,000,0

Summary

Log Pair USD

Local CCY	Local Side	Value Date	Est Local Amount	Leg Side	Est Leg Amount	WM/Refinitiv
-----------	------------	------------	------------------	----------	----------------	--------------

A Foundation for a Full Solution

The proposal supports the current FDC3 API...

...but theres more to do!

- The lack of identity / auth features in the FDC3 API is a problem
 - User identity: Is the same user operating each Desktop Agent?
 - App identity: Do we want to interoperate with every other app?
 - Desktop Agent Identity: Do we trust all of the Desktop Agents relaying messages?
- Bridging supports auth for Desktop Agents
- Simplifying assumption/restriction applied:
 - Connecting to localhost + cross-machine interop as an internal concern of the bridge
- Bridging will/must adopt whatever solutions are chosen for the FDC3 API.

@experimental

Bridging is proposed with the @experimental designation

@experimental was added to FDC3's governance in 2.0...

- ...to support features with a tentative design,
- ...that may then be refined via feedback,
- ...before their inclusion in FDC3 is finalized.
 - Implies an optional compliance criteria until finalized

Easier for multiple stakeholders to agree on, implement and report back on as an @experimental part of the Standard

What's the impact on FDC3?

- Hardly any change to the FDC3 API (low/no impact on App developers)
 - A set of recommended channels (DONE!),
 - One field in an AppIdentifier (to target/attribute messages to a Desktop Agent)
 - 2 error messages (in existing places)
- Simplifies complex situations for FDC3 adopters
 - To enable the next round of FDC3 uptake!
- Allows FDC3 to more fully realise its goals/charter
 - Integration between applications without prior bilateral (or multilateral!) agreements
- Creates an alternative route to integration for large/monolithic native apps (and their vendors!)

What's the impact on FDC3?

- Hardly any change to the FDC3 API (low/no impact on App developers)
 - A set of recommended channels (DONE!),
 - One field in an AppIdentifier (to +
 - 2 error messages

▪ Simplified

Why?

- It's easier to agree on working in an open standard
- Less throwaway customisation
- Cross platform/vendor/technology support

▪ And

▪

applications without prior bilateral (or multilateral!)

- An alternative route to integration for large/monolithic native apps (and their vendors!)

Next Steps

- Multiple implementations underway
 - Backplane seeking contributors and adopters
- Review the PR
 - PR: <https://github.com/finos/FDC3/pull/968>
 - Web preview:
<https://deploy-preview-968--fdc3.netlify.app/docs/next/agent-bridging/spec>
- Voting on inclusion in FDC3 or modifying it to exist outside FDC3 as a separate standard (FDC3 maintainers to agree exact wording)
 - SWG meeting Thursday 22nd June 2023
 - Vote to be opened via email for *registered* voting participants



FINOS

Fintech
Open Source
Foundation



FDC³

Thank you!

finos.org