FINOS

Fintech
Open Source
Foundation

FDC3

Standards Working Group

23rd June 2022

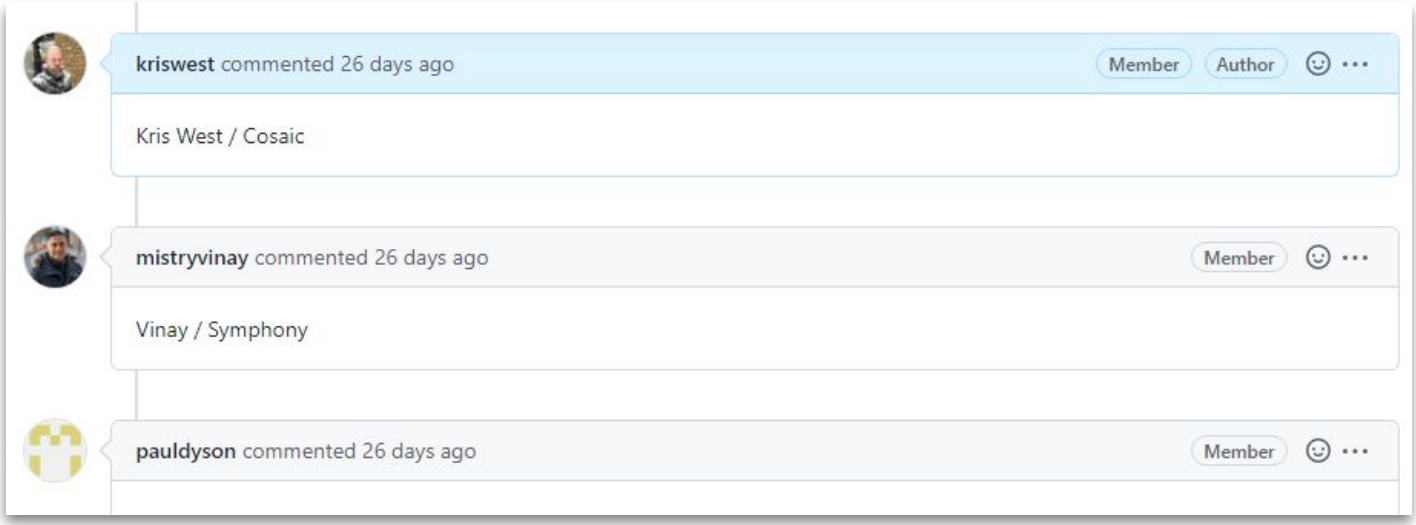
FDC3 2.0 adoption

Meeting notices

- FINOS Project leads are responsible for observing the FINOS guidelines for running project meetings. Project maintainers can find additional resources in the FINOS Maintainers Cheatsheet.
- All participants in FINOS project meetings are subject to the LF Antitrust Policy, the FINOS Community Code of Conduct and all other FINOS policies.
- FINOS meetings involve participation by industry competitors, and it is the intention of FINOS and the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws. Please contact legal@finos.org with any questions.
- **FINOS project meetings may be recorded for use solely by the FINOS team for administration purposes. In very limited instances, and with explicit approval, recordings may be made more widely available.**
    - **The recording of today's presentation section (but not the Q&A session) will be published and shared. Please hold your questions until the end of the session**

Please record your meeting attendance

https://github.com/finos/FDC3/issues/757



Agenda

- Convene & roll call, review meeting notices (5mins)
- Review action items from previous meeting (5mins)
  - Standard WG Meeting - May 26, 2022 #728
- FDC3 2.0 Adoption vote details (20 mins)
  - Summary of changes in FDC3 2.0
    - See pre-draft at https://fdc3.finos.org/docs/next/fdc3-intro
  - How to vote on adoption
  - Release details
- FDC3 2.0 Q&A session (20 mins)
- Call for roadmap items for FDC3 2.1 and beyond (5mins)
  - Existing issues review
- AOB & Adjourn (5mins)

FDC3 2.0

Adoption vote details

Summary of changes in FDC3 2.0

Discussion groups & work streams:

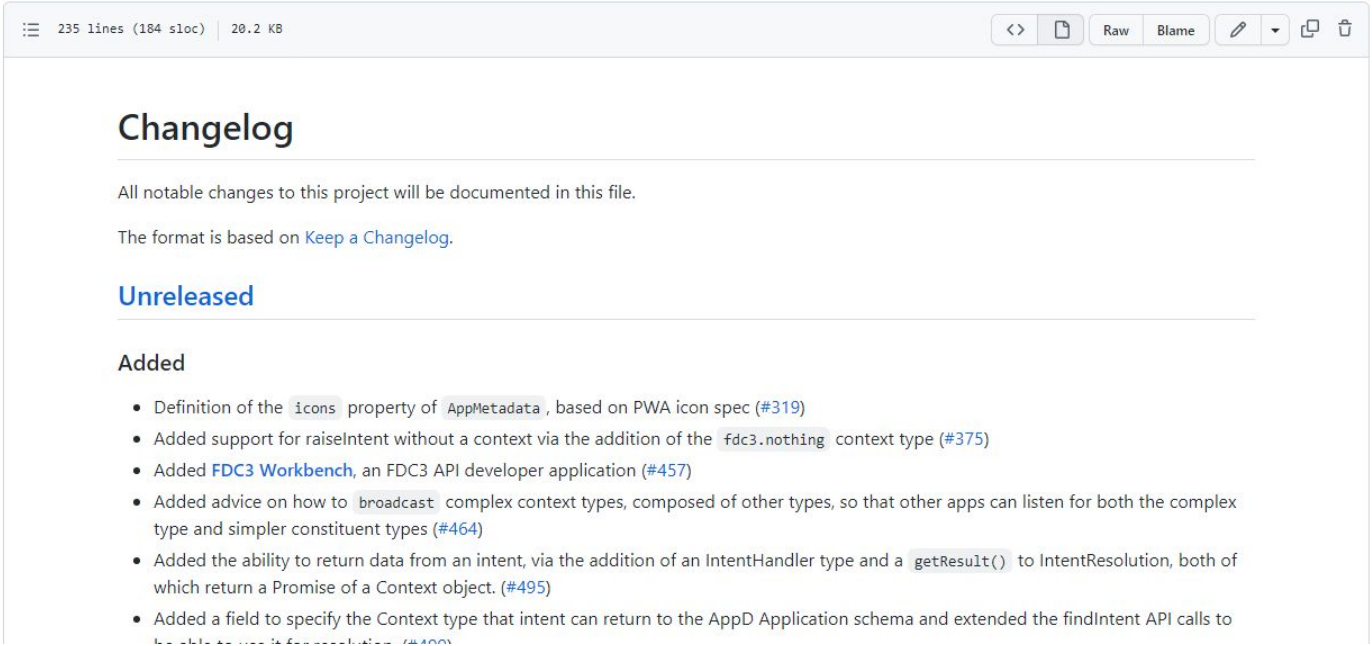
- api Refining the FDC3 API (continuing on from channels feeds & transactions)
- Context Data & Intents Proposing new content types and intents
- app-directory Proposing improvements to AppD
- formal specification Improving the standard for the FDC3 docs and website

On-going work streams for future releases:

- Context Data & Intents Proposing new content types and intents
- Desktop Agent Bridging Enabling interop between different desktop agents
- conformance Desktop Agent API conformance testing framework for FDC3

FDC3 2.0 Changelog

https://github.com/finos/FDC3/blob/master/CHANGELOG.md

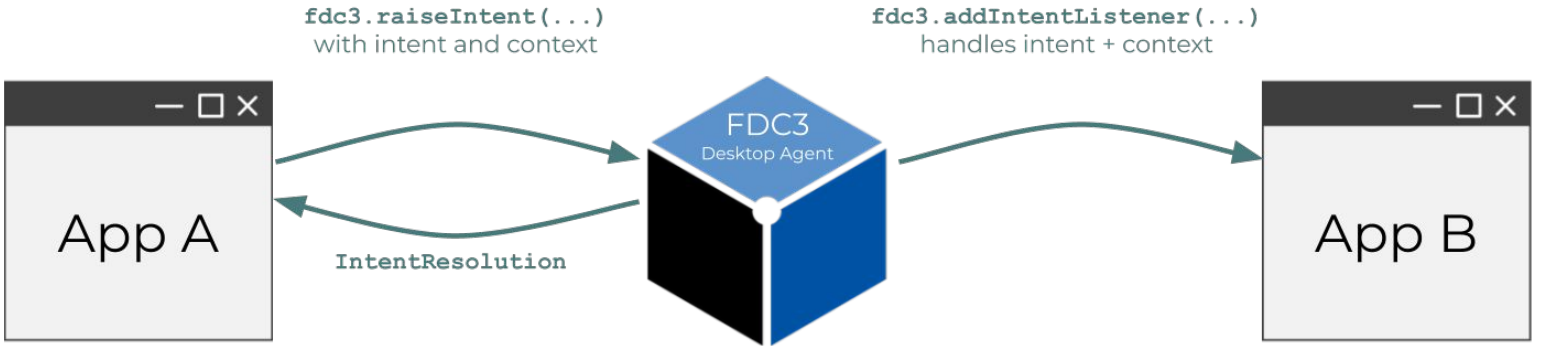


Desktop Agent API

api channels feeds & transactions

What's new?

- **Expanded intent handling to support data exchange**
  - FDC3 v1.* allows you to send requests to other applications

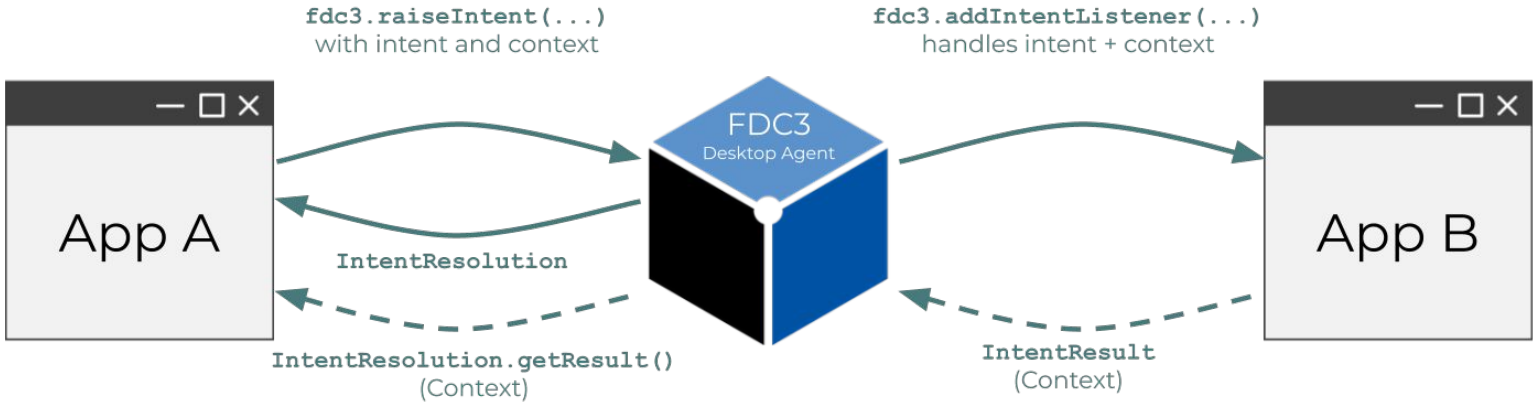


  - But building workflows that involve retrieving data from other applications is hard

Desktop Agent API

api | channels feeds & transactions

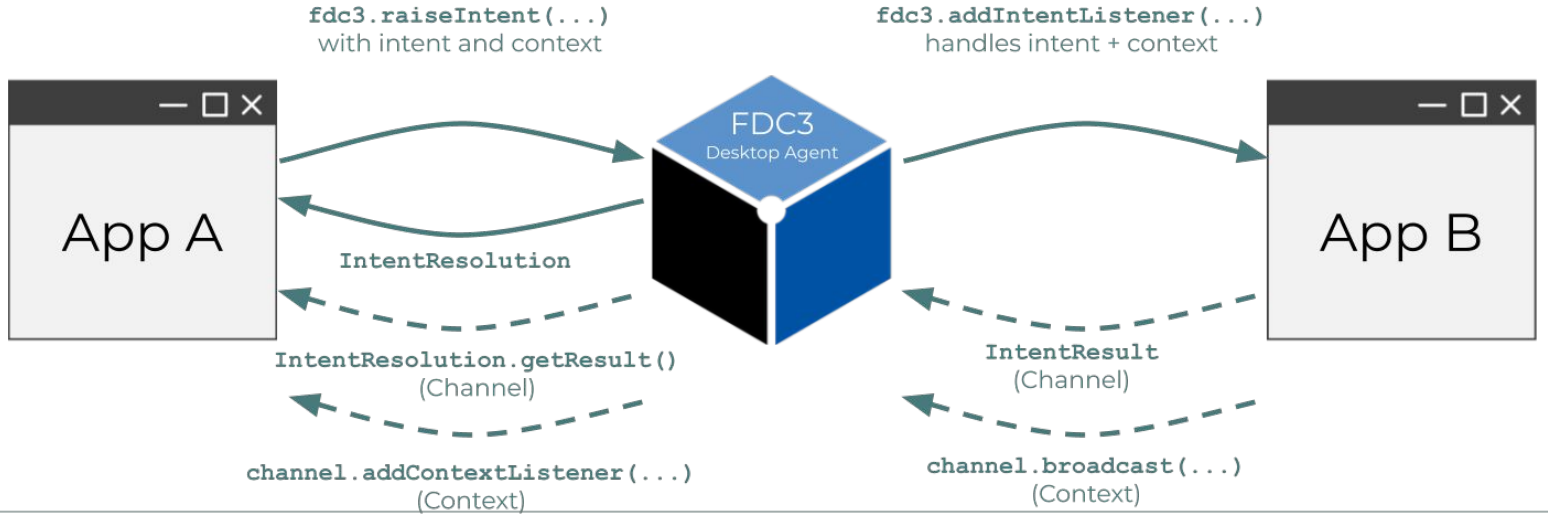
- **Transactions** -> retrieve a Context as an **`IntentResult`** from a raised Intent
  - Always intended, but never fully possible



Desktop Agent API

api | channels feeds & transactions

- **Feeds** -> retrieve a Channel as an **`IntentResult`** from a raised intent
  - Use a **`PrivateChannel`** to synchronize starting a feed, control it, or clean up after the other side leaves



Desktop Agent API

api channels feeds & transactions

- Enabling **real data exchange** allows FDC3 to tackle new use cases, such as:
  - Create real APIs with **transactions**, e.g.
    - Create an order and receive back the id and other details.
    - Create, update or retrieve a profile and receive the (new, merged or retrieved) record.
    - Start a Chat a receive back a reference to the room to send further messages.
  - Request data **feeds**, e.g.
    - A pricing stream for a specified instrument.
    - A stream of trades to be plotted on a chart as they happen.
    - A stream of order details or updates..

Desktop Agent API

api channels feeds & transactions

- Made FDC3 aware of and able to use **identities for individual instances of apps**
  - Optional **`instanceId`** property of **`AppIdentifier`**
  - Enumerate or otherwise receive the identities of existing instances of applications and target new intents
- (Optional) ability to expose the **originating app metadata** for context or intent messages
  - Lets you know who you are interoperating with (including instance details)
- (Recommended) **Standard set of user channels**
  - Will be important for future revisions where bridging of multiple desktop agents may become possible

Desktop Agent API

api channels feeds & transactions

- Cleaning up the API
  - Standardize on **`appId`** (rather than **`name`**) to identify an app
    - Passed to and received from the API via the new **`AppIdentifer`** type, which is a parent type / subset of **`AppMetadata`** and used in preference to it)
    - **`appId`** is required, **`name`** is optional
    - Deprecated function signatures based on **`name`**
  - Numerous **clarifications on API use.**
  - **Better errors** and documentation of them.
  - Standardizing on **async function calls.**
  - '**System channels' renamed 'User channels**' to clarify usage.
  - **Removed the 'global' channel** (deprecated in an earlier version).

App Directory

app-directory

What's new?

- Reconfirmed (and better documented) its **role and use-case in FDC3**
- Updated both the API and application records with the goal of making them **'fit for purpose' for the next few years**
    - Which is critical given FDC3 is achieving strong uptake from industry at the moment!
- **'Trimmed' the API calls down** to what's necessary or useful to standardize
    - With the assumption that a Desktop Agent could/should connect to multiple app directories

App Directory

app-directory

- Significantly Improved the application records:
  - Support for adding or linking **manifests for multiple containers**
  - Moving towards **container-independence** (vendor-agnostic launch **`details`**)
  - Added **`category`** metadata data for **app browsing and search use cases**
  - Better **support for 'native' apps** (app **`type`** and launch **`details`**)
  - Better **aligned with web application manifest**
  - Support for **localisation and accessibility** (translations and language tags for screen readers)
  - **Describe how your app uses interop** (for developer support and to allow browsing interfaces to help you find apps to interop with)

Context Data & Intents

What's new?

- Many **new app use cases are supported** with new or improved standard context types & intents:
  - Communicate: **`StartCall`**, **`StartChat`**, **`StartEmail`**
  - **Visualize:** Provide detailed config to charts
  - Research: **`ViewResearch`** intent
  - Interact with a CRM: **`ViewProfile`** (replaces **`ViewContact`**), **`ViewInteractions`**, **`StartCall`**, **`StartChat`**, **`StartEmail`**
  - Interact with an OMS: **`ViewQuotes`**, **`ViewOrders`**

Context Data & Intents

- **Improved building blocks** that will be used to create future intents & contexts
  - **Context Field type conventions** (e.g. id, date, currency, country etc.)
  - Expanded **Intent name conventions**
    - To promote consistency and to make the resulting actions intuitive for users
    - Added conventions for intents that create, update or retrieve data
  - Context types: **`TimeRange`**, **`Currency`**, **`Country`**, **`Valuation`**

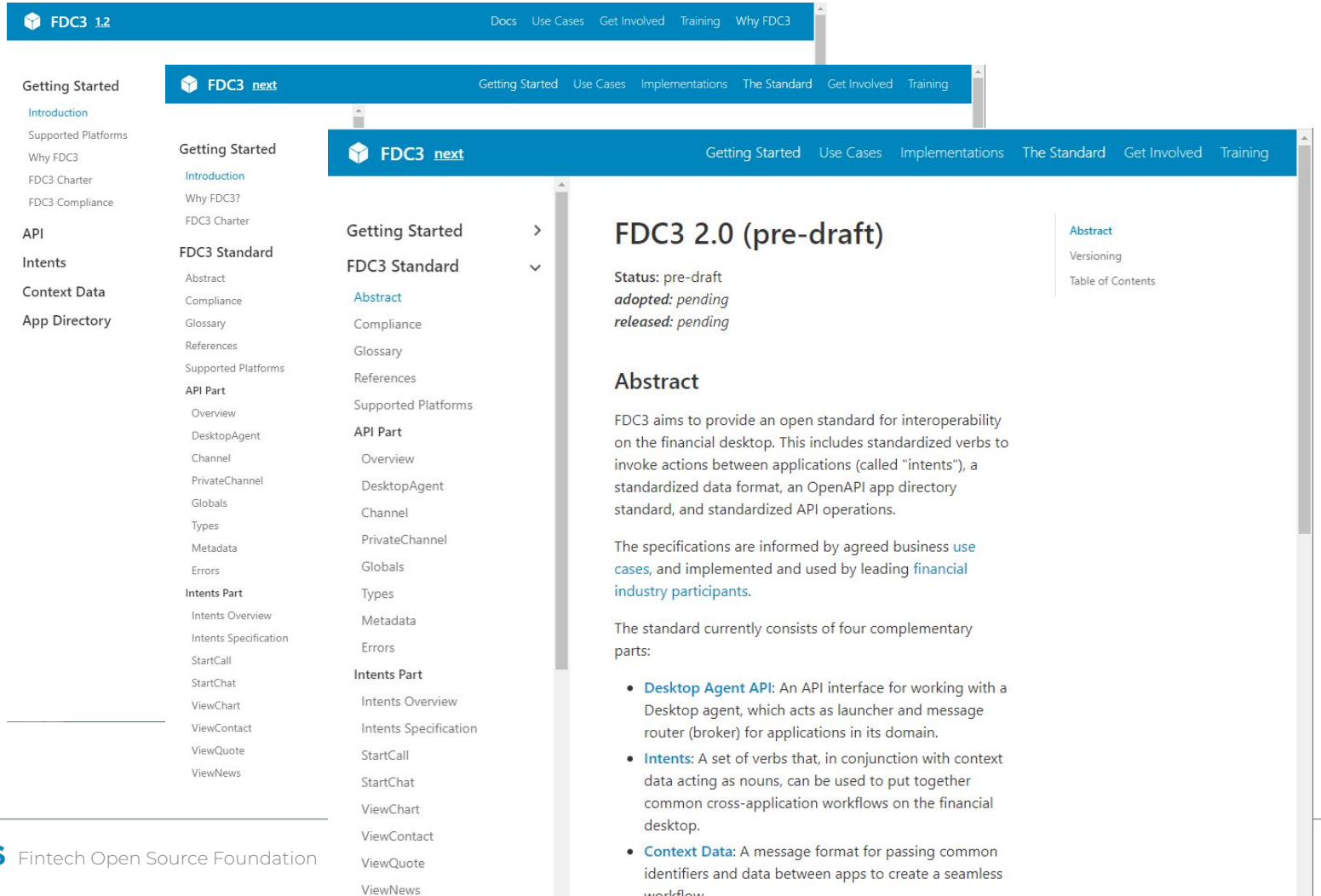
Improving the FDC3 Specification(s)

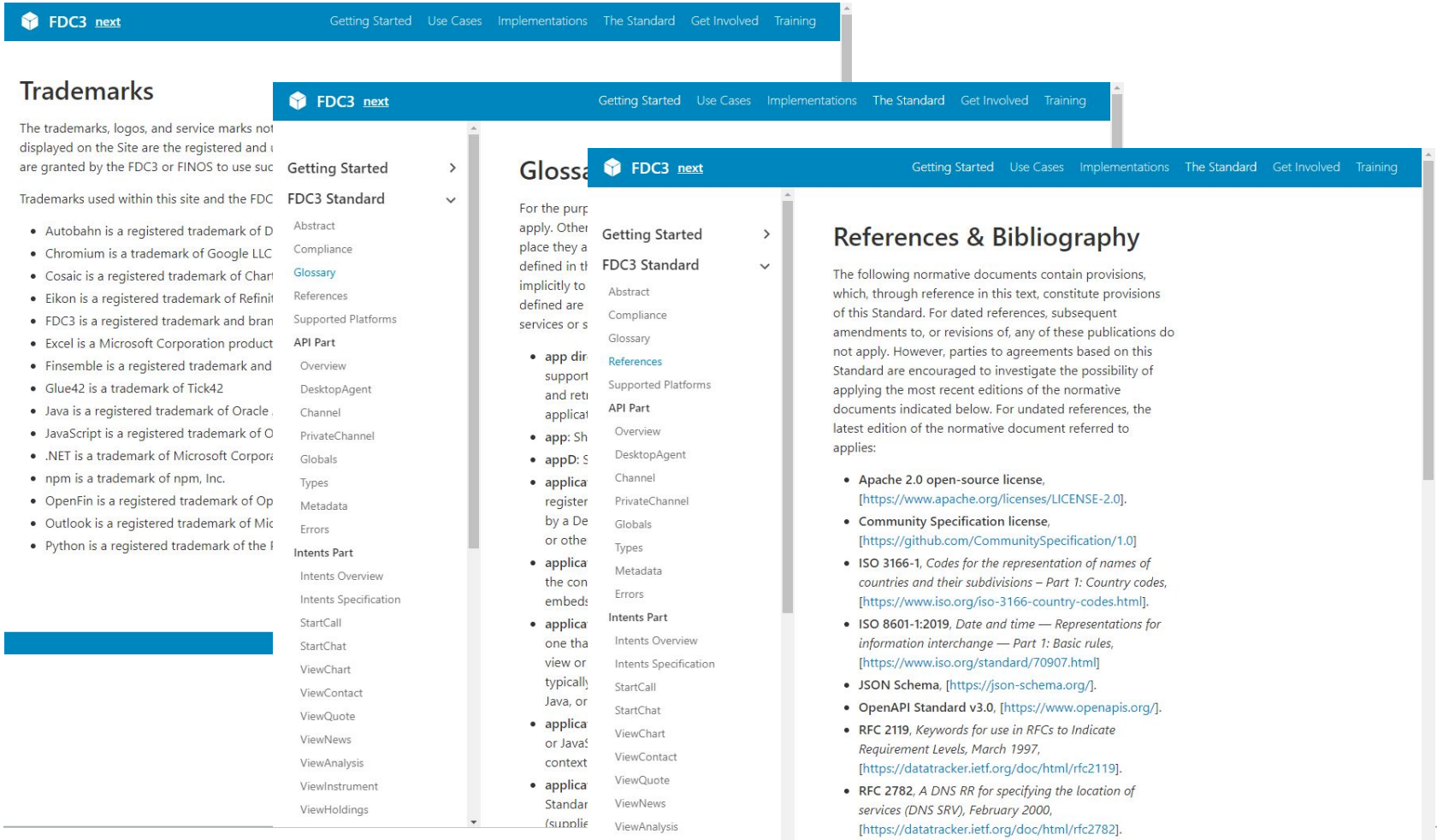
formal specification

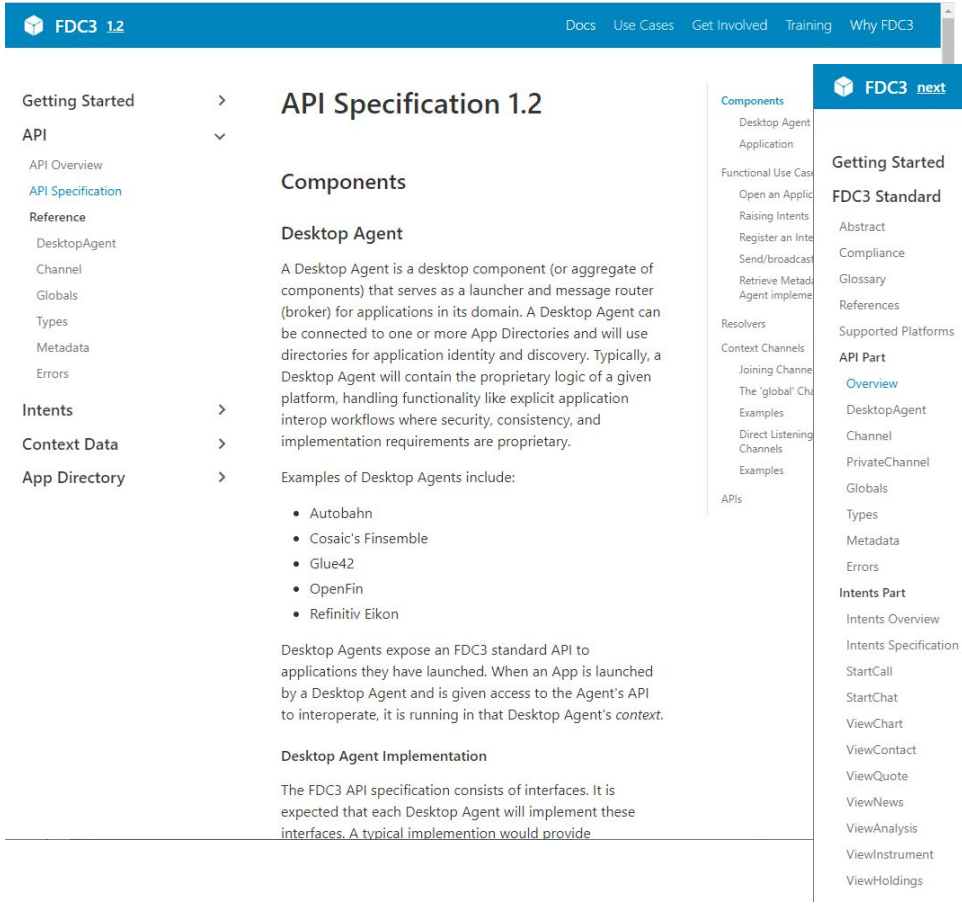
FINOS-supported project to **improve and formalize the documentation of the FDC3 standard** which delivered changes that:

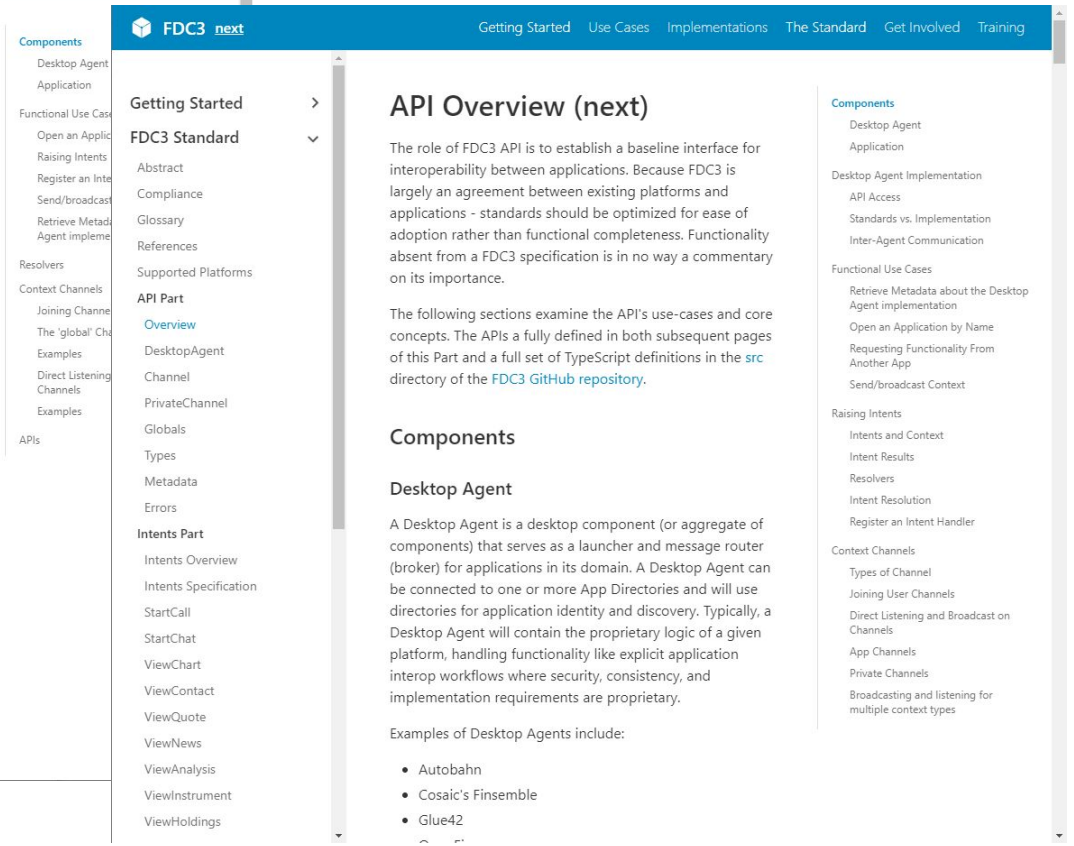
- Better integrate the separate parts of FDC3 into a single Standard.
- Consolidate and improve the docs for each 'part' of the Standard.
  - Better define and use terminology, backed by a glossary.
  - Clarify compliance requirements.
  - Deduplicate docs and improve their structure
- Reference external standards and trademarks.
- Better document governance procedures and policies.

Rex Jaeschke
Consultant editor


FDC3 1.2
Docs
Use Cases
Get Involved
Training
Why FDC3
Getting Started
Introduction
Supported Platforms
Why FDC3
FDC3 Charter
FDC3 Compliance
API
Intents
Context Data
App Directory
FDC3 next
Getting Started
Use Cases
Implementations
The Standard
Get Involved
Training
Getting Started
Introduction
Why FDC3?
FDC3 Charter
FDC3 Standard
Abstract
Compliance
Glossary
References
Supported Platforms
API Part
Overview
DesktopAgent
Channel
PrivateChannel
Globals
Types
Metadata
Errors
Intents Part
Intents Overview
Intents Specification
StartCall
StartChat
ViewChart
ViewContact
ViewQuote
ViewNews
FDC3 next
Getting Started
Use Cases
Implementations
The Standard
Get Involved
Training
Getting Started
FDC3 Standard
Abstract
Compliance
Glossary
References
Supported Platforms
API Part
Overview
DesktopAgent
Channel
PrivateChannel
Globals
Types
Metadata
Errors
Intents Part
Intents Overview
Intents Specification
StartCall
StartChat
ViewChart
ViewContact
ViewQuote
ViewNews
FDC3 2.0 (pre-draft)
Status: pre-draft
adopted: pending
released: pending
Abstract
Versioning
Table of Contents
Abstract
FDC3 aims to provide an open standard for interoperability on the financial desktop. This includes standardized verbs to invoke actions between applications (called "intents"), a standardized data format, an OpenAPI app directory standard, and standardized API operations.
The specifications are informed by agreed business use cases, and implemented and used by leading financial industry participants.
The standard currently consists of four complementary parts:
Desktop Agent API: An API interface for working with a Desktop agent, which acts as launcher and message router (broker) for applications in its domain.
Intents: A set of verbs that, in conjunction with context data acting as nouns, can be used to put together common cross-application workflows on the financial desktop.
Context Data: A message format for passing common identifiers and data between apps to create a seamless


FDC3 next
Getting Started
Use Cases
Implementations
The Standard
Get Involved
Training
Trademarks
The trademarks, logos, and service marks not
displayed on the Site are the registered and
are granted by the FDC3 or FINOS to use suc
Trademarks used within this site and the FDC
Autobahn is a registered trademark of D
Chromium is a trademark of Google LLC
Cosaic is a registered trademark of Chart
Eikon is a registered trademark of Refinit
FDC3 is a registered trademark and bran
Excel is a Microsoft Corporation product
Finsemble is a registered trademark and
Glue42 is a trademark of Tick42
Java is a registered trademark of Oracle
JavaScript is a registered trademark of O
.NET is a trademark of Microsoft Corpor
npm is a trademark of npm, Inc.
OpenFin is a registered trademark of Op
Outlook is a registered trademark of Mic
Python is a registered trademark of the
FDC3 next
Getting Started
Use Cases
Implementations
The Standard
Get Involved
Training
Getting Started
FDC3 Standard
Abstract
Compliance
Glossary
References
Supported Platforms
API Part
Overview
DesktopAgent
Channel
PrivateChannel
Globals
Types
Metadata
Errors
Intents Part
Intents Overview
Intents Specification
StartCall
StartChat
ViewChart
ViewContact
ViewQuote
ViewNews
ViewAnalysis
ViewInstrument
ViewHoldings
Glossa
For the purp
apply. Other
place they a
defined in th
implicitly to
defined are
services or s
app dir
support
and retu
applicat
app: Sh
appD: S
applica
register
by a De
or othe
applica
the con
embeds
applica
one tha
view or
typically
Java, or
applica
or Java
context
applica
Standar
(supplie
FDC3 next
Getting Started
Use Cases
Implementations
The Standard
Get Involved
Training
Getting Started
FDC3 Standard
Abstract
Compliance
Glossary
References
Supported Platforms
API Part
Overview
DesktopAgent
Channel
PrivateChannel
Globals
Types
Metadata
Errors
Intents Part
Intents Overview
Intents Specification
StartCall
StartChat
ViewChart
ViewContact
ViewQuote
ViewNews
ViewAnalysis
ViewInstrument
ViewHoldings
References & Bibliography
The following normative documents contain provisions, which, through reference in this text, constitute provisions of this Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies:
Apache 2.0 open-source license, [https://www.apache.org/licenses/LICENSE-2.0].
Community Specification license, [https://github.com/CommunitySpecification/1.0]
ISO 3166-1, Codes for the representation of names of countries and their subdivisions – Part 1: Country codes, [https://www.iso.org/iso-3166-country-codes.html].
ISO 8601-1:2019, Date and time — Representations for information interchange — Part 1: Basic rules, [https://www.iso.org/standard/70907.html]
JSON Schema, [https://json-schema.org/].
OpenAPI Standard v3.0, [https://www.openapis.org/].
RFC 2119, Keywords for use in RFCs to Indicate Requirement Levels, March 1997, [https://datatracker.ietf.org/doc/html/rfc2119].
RFC 2782, A DNS RR for specifying the location of services (DNS SRV), February 2000, [https://datatracker.ietf.org/doc/html/rfc2782].
TypeScript Programming Language, [https://www.typescriptlang.org/].


FDC3 1.2
Docs
Use Cases
Get Involved
Training
Why FDC3
Getting Started
API
API Overview
API Specification
Reference
DesktopAgent
Channel
Globals
Types
Metadata
Errors
Intents
Context Data
App Directory
API Specification 1.2
Components
Desktop Agent
A Desktop Agent is a desktop component (or aggregate of components) that serves as a launcher and message router (broker) for applications in its domain. A Desktop Agent can be connected to one or more App Directories and will use directories for application identity and discovery. Typically, a Desktop Agent will contain the proprietary logic of a given platform, handling functionality like explicit application interop workflows where security, consistency, and implementation requirements are proprietary.
Examples of Desktop Agents include:
Autobahn
Cosaic's Finsemble
Glue42
OpenFin
Refinitiv Eikon
Desktop Agents expose an FDC3 standard API to applications they have launched. When an App is launched by a Desktop Agent and is given access to the Agent's API to interoperate, it is running in that Desktop Agent's context.
Desktop Agent Implementation
The FDC3 API specification consists of interfaces. It is expected that each Desktop Agent will implement these interfaces. A typical implemention would provide


FDC3 next
Getting Started
Use Cases
Implementations
The Standard
Get Involved
Training
Getting Started
FDC3 Standard
Abstract
Compliance
Glossary
References
Supported Platforms
API Part
Overview
DesktopAgent
Channel
PrivateChannel
Globals
Types
Metadata
Errors
Intents Part
Intents Overview
Intents Specification
StartCall
StartChat
ViewChart
ViewContact
ViewQuote
ViewNews
ViewAnalysis
ViewInstrument
ViewHoldings
API Overview (next)
The role of FDC3 API is to establish a baseline interface for interoperability between applications. Because FDC3 is largely an agreement between existing platforms and applications - standards should be optimized for ease of adoption rather than functional completeness. Functionality absent from a FDC3 specification is in no way a commentary on its importance.
The following sections examine the API's use-cases and core concepts. The APIs a fully defined in both subsequent pages of this Part and a full set of TypeScript definitions in the src directory of the FDC3 GitHub repository.
Components
Desktop Agent
A Desktop Agent is a desktop component (or aggregate of components) that serves as a launcher and message router (broker) for applications in its domain. A Desktop Agent can be connected to one or more App Directories and will use directories for application identity and discovery. Typically, a Desktop Agent will contain the proprietary logic of a given platform, handling functionality like explicit application interop workflows where security, consistency, and implementation requirements are proprietary.
Examples of Desktop Agents include:
Autobahn
Cosaic's Finsemble
Glue42
Components
Desktop Agent
Application
Desktop Agent Implementation
API Access
Standards vs. Implementation
Inter-Agent Communication
Functional Use Cases
Retrieve Metadata about the Desktop Agent implementation
Open an Application by Name
Requesting Functionality From Another App
Send/broadcast Context
Raising Intents
Intents and Context
Intent Results
Resolvers
Intent Resolution
Register an Intent Handler
Context Channels
Types of Channel
Joining User Channels
Direct Listening and Broadcast on Channels
App Channels
Private Channels
Broadcasting and listening for multiple context types

How to vote on adoption

Voting is open to **enrolled** FDC3 voting participants

If you plan to use, implement or benefit from the FDC3 Standard, FINOS asks you to sign up and enroll as a voting participant.

To enroll as a voting participant send an email:

- **to:** fdc3-participants+subscribe@finos.org
- **subject line:** Please enroll me as FDC3 standard participant
- **body:**
  Hi, my name is, ____, and I'd like to formally participate to the FDC3 standard process. I plan to contribute on behalf of ___ . Thank you!

Note:

- Standard participants are bound to the provisions in the FINOS IP Policy as described in the FINOS Standards governance document.
- Upon enrollment as an FDC3 voting participant, you will be invited to join the FINOS GitHub organization and the fdc3-participants GitHub team.

How to vote on adoption

1. Look for email from fdc3-participants@finos.org mailing list: "Voting open: FDC3 2.0 Standard"
2. Reply with:
   - **+1** to indicate **acceptance**
     or
   - **-1** (and an explanatory comment) to indicate **rejection**
3. Maintainers will attempt to work with anyone voting -1 to clarify changes or otherwise work through issues.

Voting opens today and will remain open until **5pm BST Thursday 30th June**.

Release details

What happens after the vote?

- **'pre-draft' promoted to 'draft' status**
- **45-day wait after adoption vote** before it is promoted to “Approved Specification” status
  - This is specified in the FINOS Standards Project Blueprint, which we are about to adopt post-2.0, makes sense to observe it this time as well,
- **Maintainers work to release**
  - Create numbered version in the website #758
  - Generate types, update the NPM module and release #745 #746
  - Merge Compliance details round-up #753
  - Add detail on any optional features in Desktop Agent API to Implementation metadata #636
  - Update Readme / introductory materials #750

Release details (2)

What happens after the vote?

- Begin work to **update FDC3 Workbench, FDC3 eXplained to 2.0**
- Finish adoption of **Community Spec License / FINOS Standards Project Blueprint**
  - Authors of open PRs to be contacted to confirm that their contributions to standard (rather than software) are made under CSL rather than Apache 2.0
- Start work on **FDC3 2.1 roadmap**

FDC3 Community page

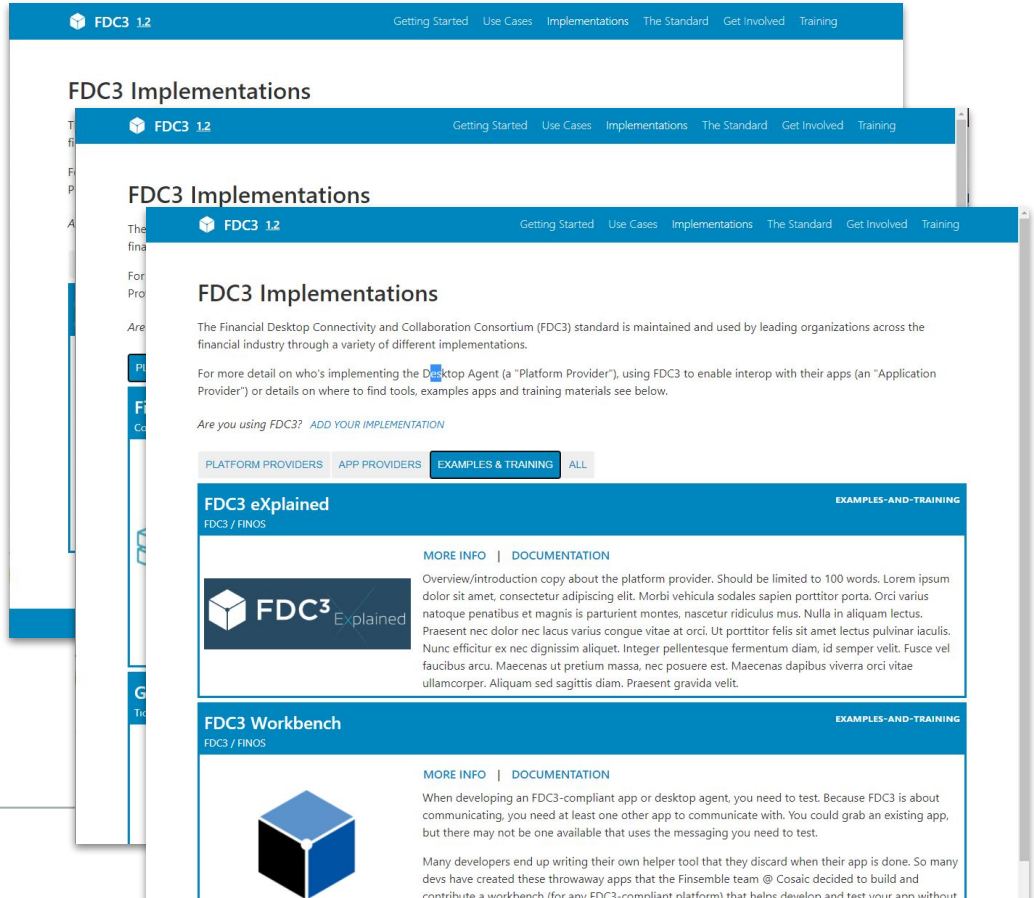
Using FDC3?

Add your details to the FDC3 website:

- Apps with FDC3 integrated*
- Desktop Agents*
- Examples, Training materials

Add in-house implementations & apps to demonstrate support for FDC3 and help grow the ecosystem!

Email a product name, 100 description, logo image and optional more info & docs links to fdc3-maintainers@finos.org or raise your own PR here



FDC3 2.0

Q & A Session

Roadmap

FDC3 2.1 and beyond

Existing open issues

25 existing open issues

Highlights:

- Desktop Agent Bridging
- Context: Order, Trade, RichTextData, improve Position & Instrument types,
- Intents: CreateInteraction, CreatOrUpdateProfile
- AppD: Add /intents and /contexts endpoints to the AppD API
- API: Provide Notification API's, Add save and restore state API calls
- API: Upgrade exposure of originating app identity to a requirement (MUST) from recommended (SHOULD)
- Identity?

Discussion Group: Desktop Agent Bridging

Desktop Agent Bridging

Purpose: To develop proposals for bridges between FDC3 Desktop Agents (DAs),, allowing apps running in differents DAs to interoperate with each other:

Status:

- Broad technical decisions made, summary:
  - Desktop Agents (DAs) will establish an authenticated websocket connection to a standalone 'bridge' service on localhost that routes requests from them to other agents. The bridge is responsible for authenticating connections, assigning DA identities, routing requests to other agents, collating responses from them and handling any timeouts,
- Join the discussion @ https://github.com/finos/FDC3/discussions/544

Next step: Initial specification draft

What else do we need to address?

Are there:

- Interfaces to your apps that you can't build with the API?
- Common use cases without standard context types or intents?
- App metadata you need but can't add to AppD records?
- Other ways that you think FDC3 can be improved?

Start a discussion: https://github.com/finos/FDC3/discussions
or raise an issue: https://github.com/finos/FDC3/issues

Questions?

Need help with using FDC3?
Unsure whether FDC3 can help solve your integration problem?

Start a discussion with the community at:
https://github.com/finos/FDC3/discussions

Or

join the discussion at the Context & Intents group meetings to discuss your use case(s)

FINOS

Fintech
Open Source
Foundation

Thank you!

finos.org