

2016-04-01 Fedora API Extensions Meeting

Date: Friday April 1 2pm EST (-5 UTC)

- Dial-in Number: (712) 775-7035
 - Participant Code: 479307#
 - International numbers: [Conference Call Information](#)
 - You may also call in using the [VoIP dialer](#) from a web browser, or Android/iOS apps
- IRC:
 - [Join the #fcrepo chat room via Freenode Web IRC](#) (enter a unique nick)
 - Or point your IRC client to #fcrepo on [irc.freenode.net](#)

Attendees

[William G. Cowan](#) - Indiana University

[Randall Floyd](#) - Indiana University

[Stefano Cossu](#) - The Art Institute of Chicago

[Ruth Duerr](#) - The Ronin Institute

[Elliot Metsger](#) - Johns Hopkins University

[Joshua Westgard](#) - University of Maryland

Meeting Goals

- Updates on progress so far

Agenda

1. LDCX report?
2. API-X patterns document from [Elliot Metsger](#) ([doc](#), [graphics](#), [thread](#))
3. Updated API-X graphics from [Stefano Cossu](#) ([graphic](#), [thread](#))
4. Vagrant + Docker for POCs (see [git repo](#) for [Aaron Birkland](#)'s experiments so far)
 - a. Status and experience so far
 - b. Relationship to testing/development/demo process in general
 - c. Other possible approaches

Minutes

- LDCX
 - notes/slides attached
 - May be software out there that we can use in core
 - Most interest seemed to be in functionality around content models
 - Better Fedora 3 disseminators
- API-X patterns doc
 - Opinionated in how it mentions patterns and documents
 - Three patterns: Filter, Direct invocation, Listener
 - Categorizes the use cases into the patterns
 - Stefano & Ruth still reviewing it
 - Aaron wants to make POCs, complementary to document
- Diagram
 - Used for LDCX, Made some concepts clear
- Vagrant + Docker POC
 - Attempt to run a demonstration environment
 - Demo/development may not use same infra
 - Is anybody using this sort of environment in production?
 - NSIDC: everything done using these technologies
 - Red/green testing, then do a swap into production
 - Looking at API-X through the lens of that process
 - Ruth: like to be able to test things as they come out, without being a developer
 - Stefano: we should provide basic services out of the box
 - most will be for community
 - Language agnostic web API
- Technologies for core
 - Java + camel? Python? Go? Erlang? Node?
 - Elliot: From a practical standpoint, Java, lots of experience
 - Another thing to consider - API, specification

- Not particularly practical at this stage
 - Create a doc for implementation
 - Core: everything that is essential for API-X
 - should be an event framework
 - Elliot: go down POC road, use that to demonstrate core
- More people comment on Elliot's patterns doc!