

Spritely for Secure Applications and Communities

Christine Lemmer-Webber, Randy Farmer, Jessica Tallon

Table of Contents

- [1. Introduction](#)
- [2. A peer-centric user environment](#)
 - [2.1. Safe applications](#)
 - [2.2. Persona management](#)
 - [2.3. Application and context management](#)
 - [2.4. Secure application cooperation](#)
 - [2.5. Local indexing](#)
- [3. Social networks, reimagined](#)
- [4. TODO Stuff to incorporate](#)
 - [4.1. Social layer](#)
 - [4.1.1. Personas, Context, Trust management](#)
 - [4.1.2. Social Communities](#)
 - [4.1.3. Discovery](#)
 - [4.1.4. Service bridging](#)
 - [4.2. Application layer](#)
 - [4.2.1. Distributed finance](#)
 - [4.2.2. Peer-to-peer user agent \("The Agency"\)](#)
 - [4.2.3. Possible demo applications](#)
- [5. Acknowledgements](#)

NOTE: This is an early draft. Has yet to be reviewed by the technical review board.

Blah blah, talk about all three papers... This paper isn't even close to being ready yet, it's just being outlined. For now, [The Heart of Spritely: Distributed Objects and Capability Security](#) is the document that needs your feedback.

1. Introduction

2. A peer-centric user environment

2.1. *Safe applications*

2.2. *Persona management*

2.3. *Application and context management*

2.4. *Secure application cooperation*

2.5. *Local indexing*

3. Social networks, reimagined

4. TODO Stuff to incorporate

4.1. *Social layer*

4.1.1. **Personas, Context, Trust management**

Requirements (assuming existing ocap system):

- Personas (Unums with user-exposed names/viewing/reference)
 - Profiles
 - General
 - Contextual
 - Notifications (big unpack)
- Petname system
- Credential system (mine the VC space)
- Requirements/ToE system
- Granted permissions tracking
- Context management (giant unpack)
- Login bridging (maybe move down)

4.1.2. **Social Communities**

- Messaging "style"
 - Ephemeral (IM, slack, signal, IRC, etc)
 - Persistent (forums, blogposts, etc)

- Relationships (social graph)
 - Types
 - Friends <->
 - Follows ->
 - ?
 - Per-persona
 - Per-context
 - Between objects (???)
 - "Communication entry paths": stamps and mutual friend, etc
- Membership (hub and spoke)
 - Social context / "group"
 - Actions / Activities
 - Create
 - Administrate
 - Remove
 - Archive / Replicate
 - Join
 - Leave
 - Invites
- Posts, threads, and artifacts (wiki)
 - All the AS2 stuff (Notes, Videos, Audio, etc)
 - Importantly, participants in the thread can help share content but participation in the thread is an essential capability, it's not ambient "what threads do you have, hmmm?"
- Content feedback and reputation
- Moderation and governance
 - Flexible policies
 - Pre-approval
 - Post-moderation
 - Flagging and review
 - Majority vote / quorum
 - Free for all
 - Bulk/AI moderation
 - (JIT) Behavior facilitation/correction semantics
 - Poster-side pre-flagging assistance
 - Governance-enforced filtering

- Analytics?! (privacy preserving?!)

"Do we use the same petname system for personas, apps, inventory, currency types, webpage bookmarks, etc etc"

There are two types of links:

- "Live object things": very behavior'y Sturdyrefs
- "File-like (updateable) data things" (Encrypted) Portable Storage
- URLs to HTTP stuff (screws up by mixing the two above things) (this is a bridge)

4.1.3. Discovery

Recognize the difference between these two:

- Kinds of information
 - "fundamentally public content"
 - you expect/want this to be available to everyone
 - you can't prevent indexing of this anyway
 - social agreement SHOULD be: everyone can copy
 - "fundamentally private content"
 - readable to only those who have permission to read
 - social agreements / community guidelines of when people can/should copy information out (are these ever machine-readable, and does the system help comply eg MarkM's "admonishment systems"?)
 - voluntary oblivious compliance auto-expiry (eg "context *requests* post removal after 30 days")
- Finding personas
- Finding services
- Search
 - Internal, within your own agency: public and private
 - Public content indexing and search
- App directory
- Sitemap-style inclusion protocol (as opposed to robots.txt as an exclusion protocol) as part of bridge

4.1.4. Service bridging

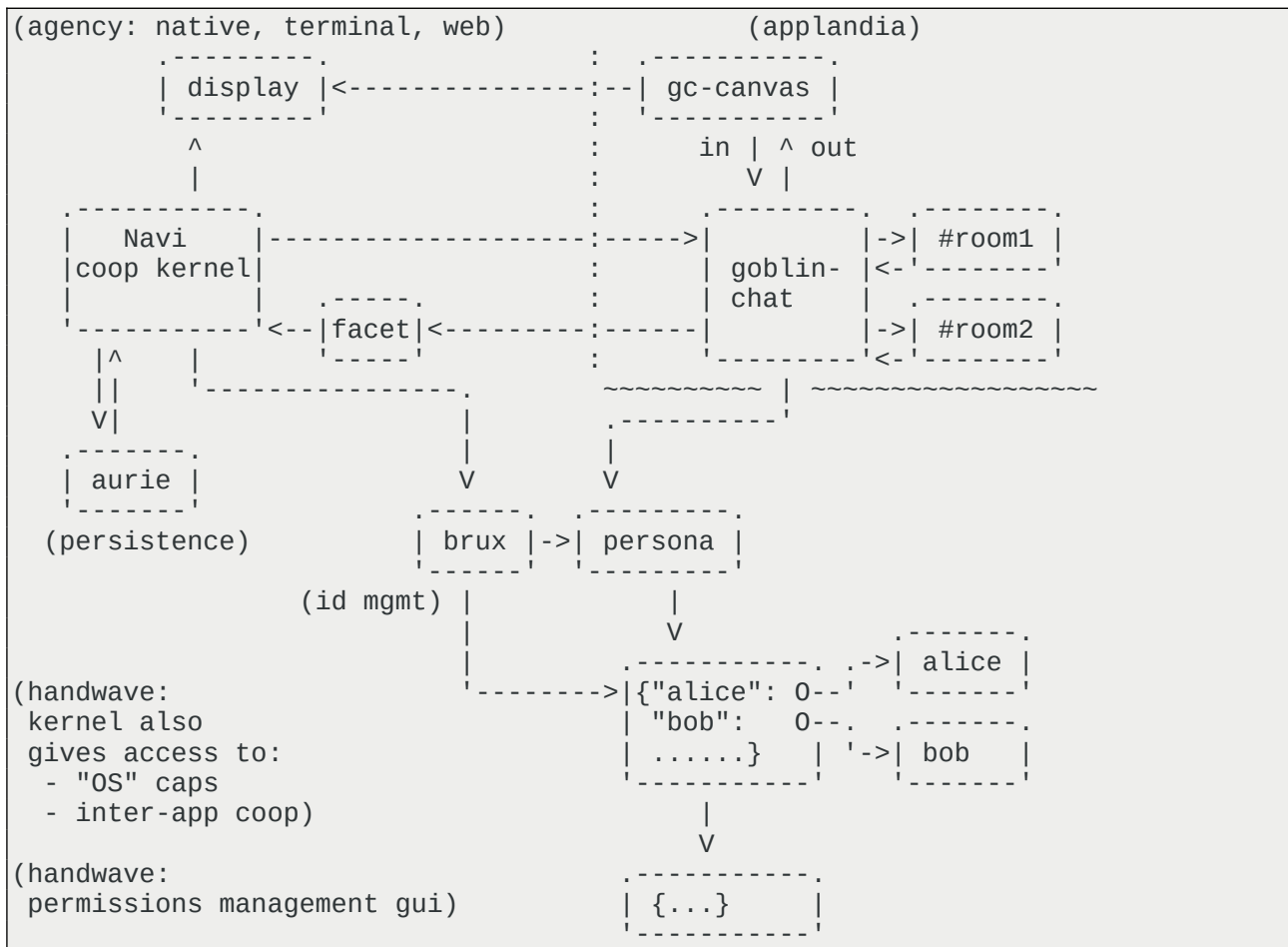
- ActivityPub
- DID Auth, whatever

4.2. Application layer

4.2.1. Distributed finance

- Agoric's stuff
 - ERTP
- Wallets blah blah

4.2.2. Peer-to-peer user agent ("The Agency")



- Inter-app cooperative kernel
 - Capability grants between apps
 - Capability grants to "OS" caps
- Permissions management gui
- Petnames integration
- Highly configurable display system
 - Per-app canvases
 - Petnames overlay tooling
 - Inter-app cooperation (with display)

- Persona profile display/management
- App persistence/storage
- Context management interface
 - Persona selection
 - App/context selection
- Flavors of the agency
 - Native
 - Linux (cross-binary support to OSX, Windows)
 - Mobile
 - Terminal
 - Web
- App install (and permissions configuration)
- Discovery integration (all kinds)
- Invitations
- Wallet/financial integration
- Bridges (and can you make it clear that things get risky here?)
 - Crossposting
 - HTML rendering of documents
 - Single-sign-on/identity mapping
- "Unum display and UI" (intentionally vague)
- Dev tools
 - MUDdebug (MUD-ish debug tools)
 - Integrated debugger
 - Minimal code editor
 - Authority scope
 - Authority of present context
 - Top-capability authority

4.2.3. Possible demo applications

- Basic social stuff
 - Ephemeral chat
 - Persistent, hostless message boards
- Convergent app examples
 - opencroquet's desktop demo,
 - collaborative text editors)

- Trading card game mockup (Sushimon)
- SpritelyMUD
- Graphical virtual world example (Liberated Pixel Cup)
- Minimal (toy?) wallet
- Introduction paths
 - Stamps
 - Friend-of-a-friend
 - Email as an introduction flow
- Streaming videoconference

5. Acknowledgements