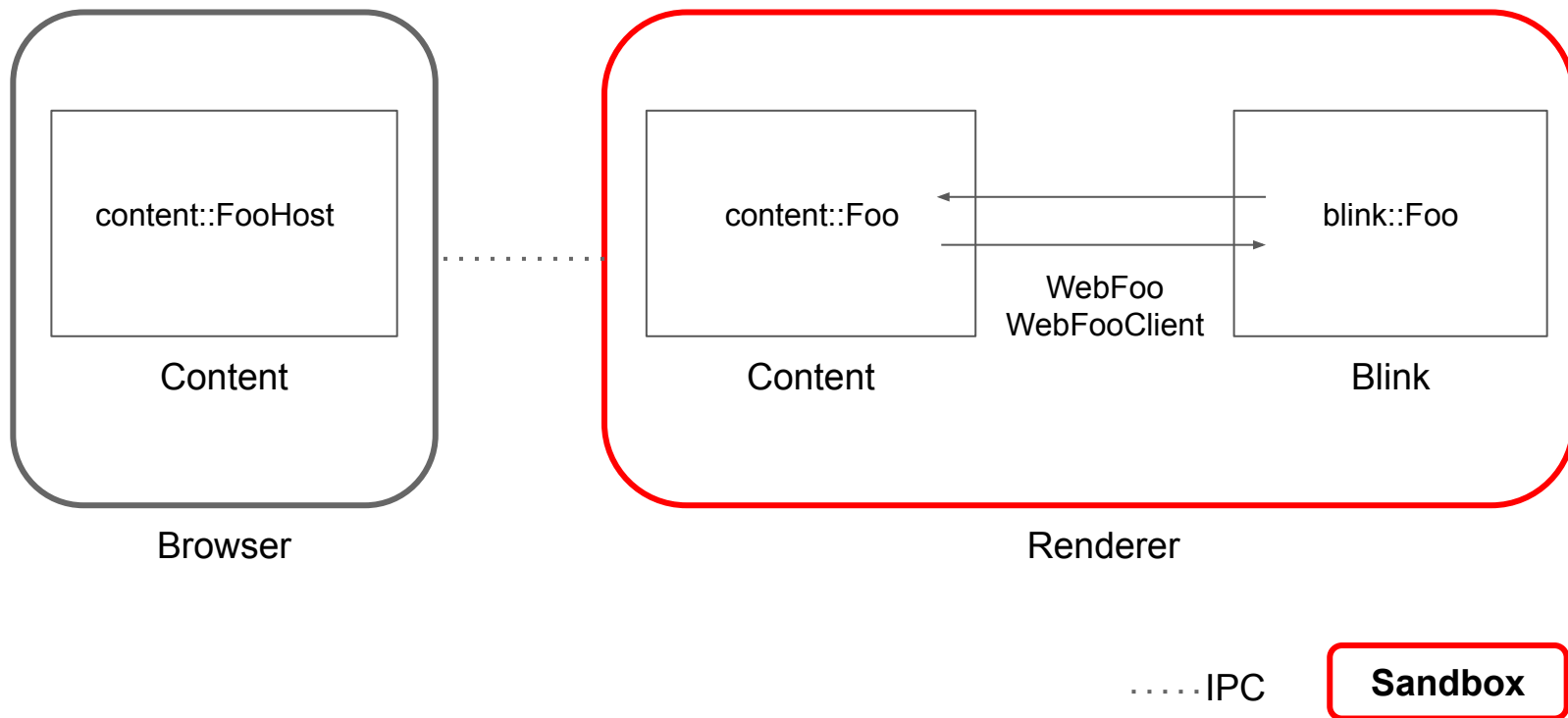# The End of Content

John Abd-El-Malek
jam@

Outline

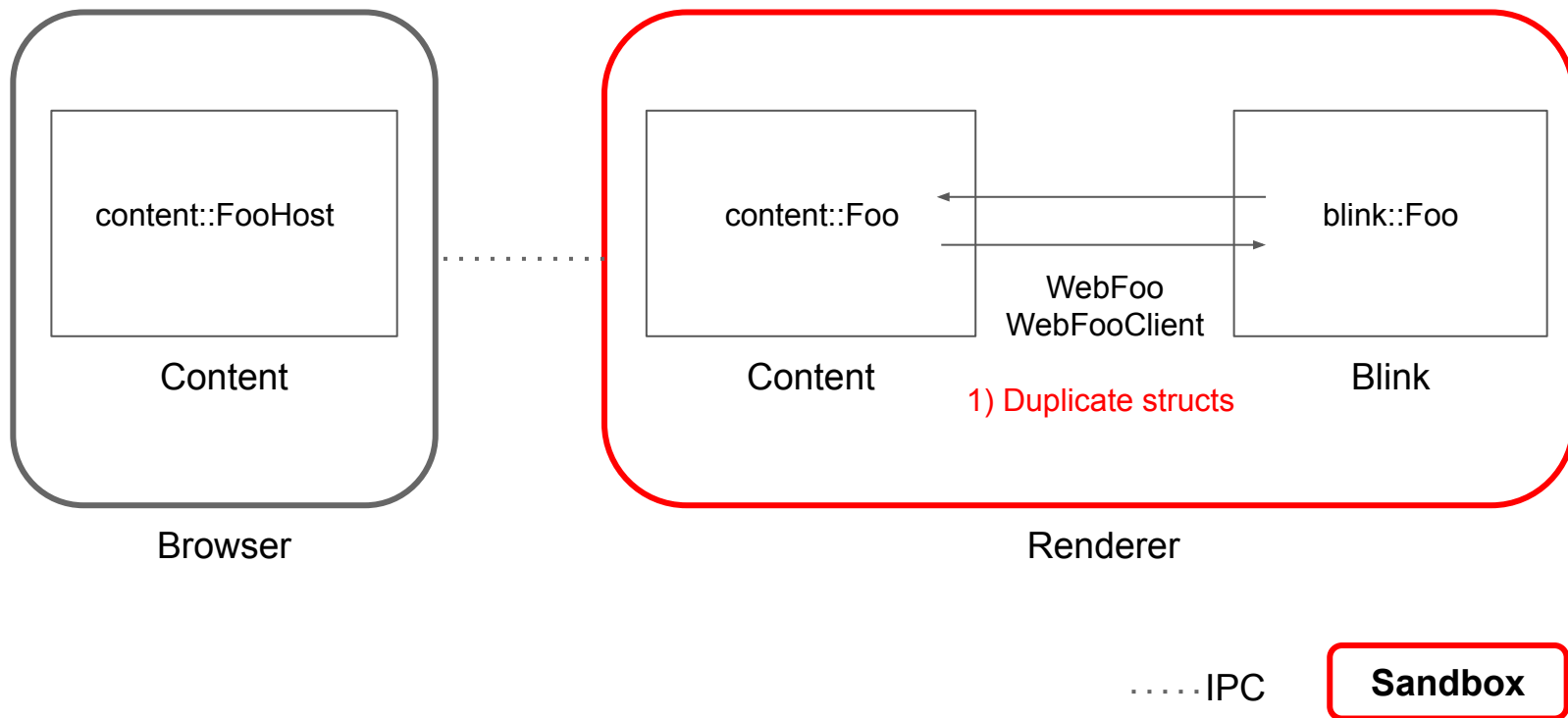1. Motivations
2. Goals
3. How
4. End State
5. Deep Dive

Motivation

- Blink originally a rendering engine
  - leaf node
  - small API to to the browser, brokered by Content which knew about IPCs
- as powerful Web App capabilities added, more features spanned processes
  - overhead
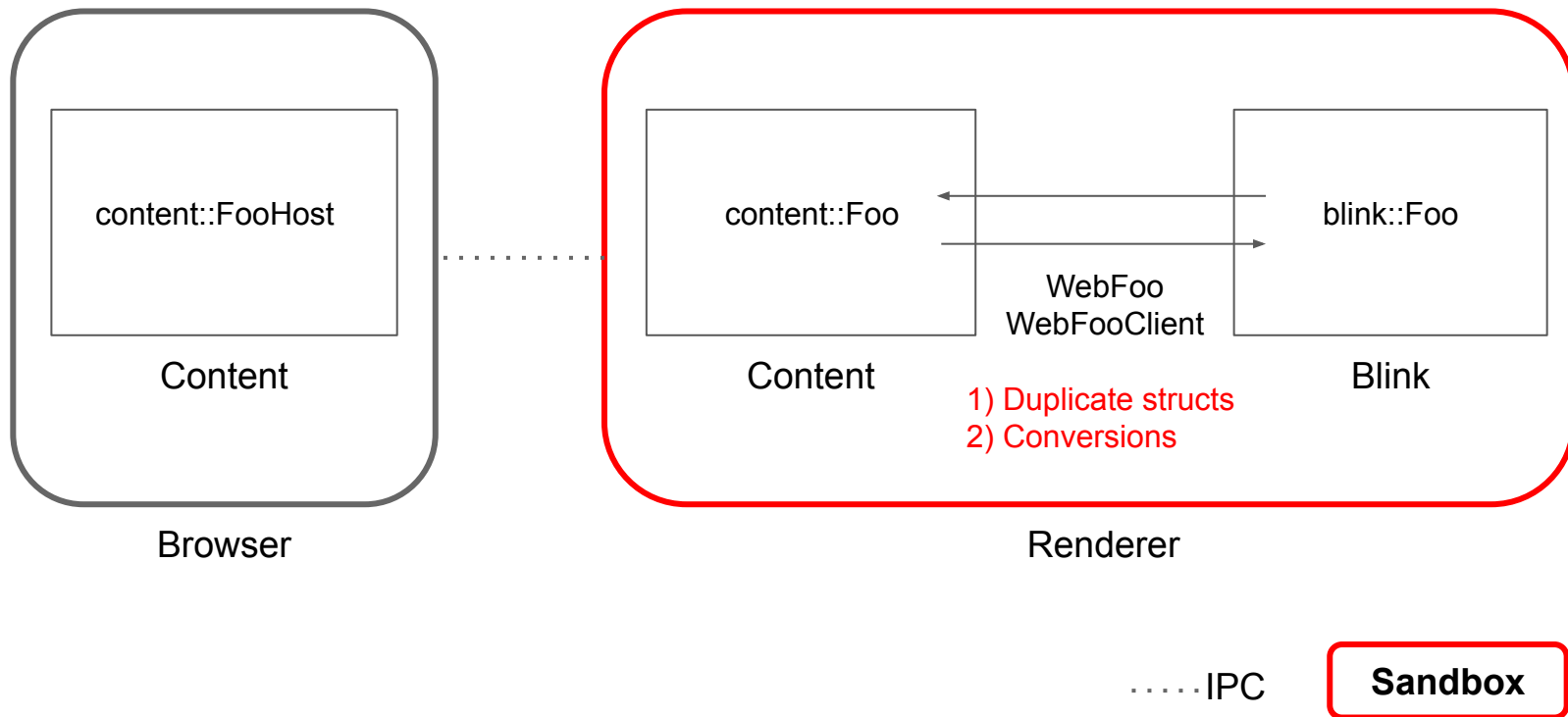 - duplication
  - performance
  - security

Motivation: Overhead

Motivation: Overhead

Motivation: Overhead

content::FooHost

Content

Browser

content::Foo

WebFoo
WebFooClient

Content

blink::Foo

Blink

1) Duplicate structs
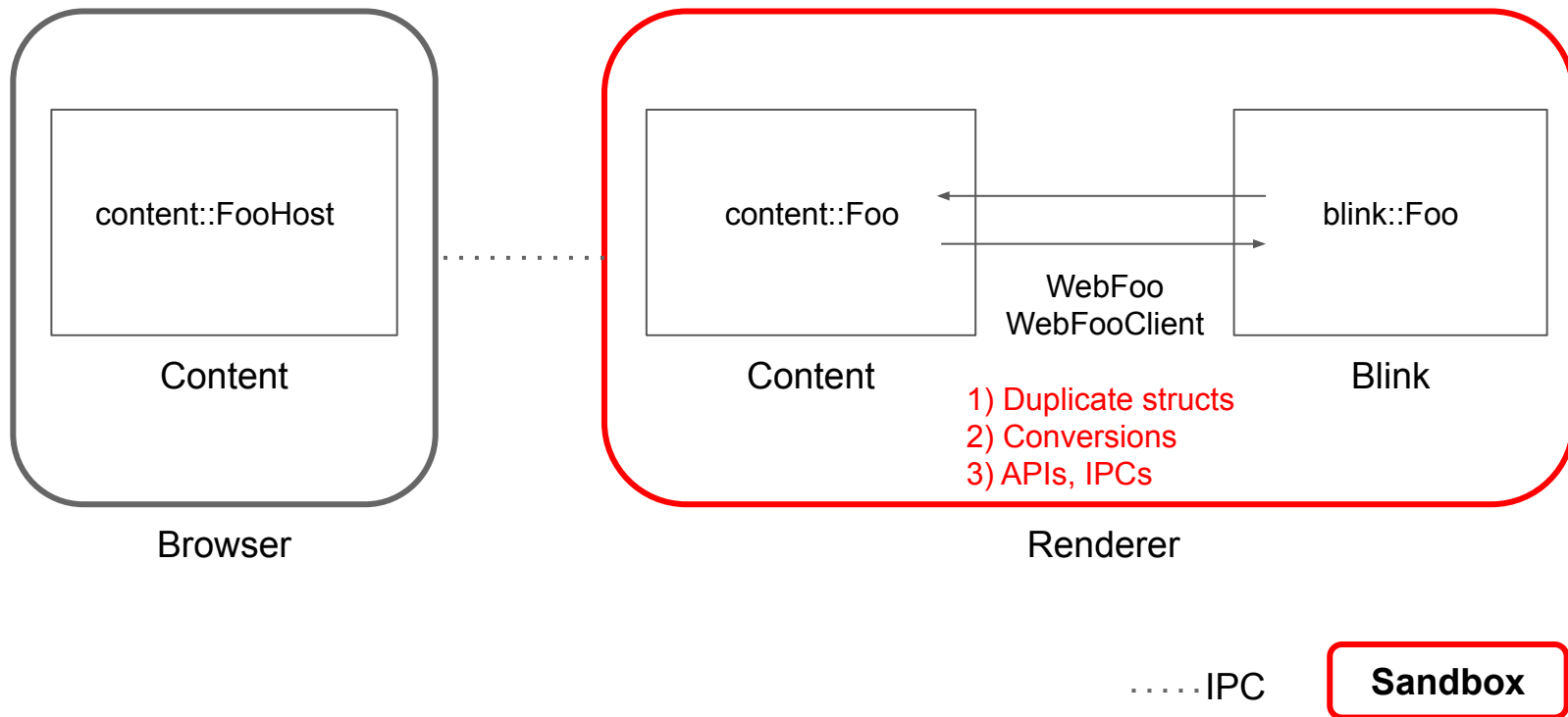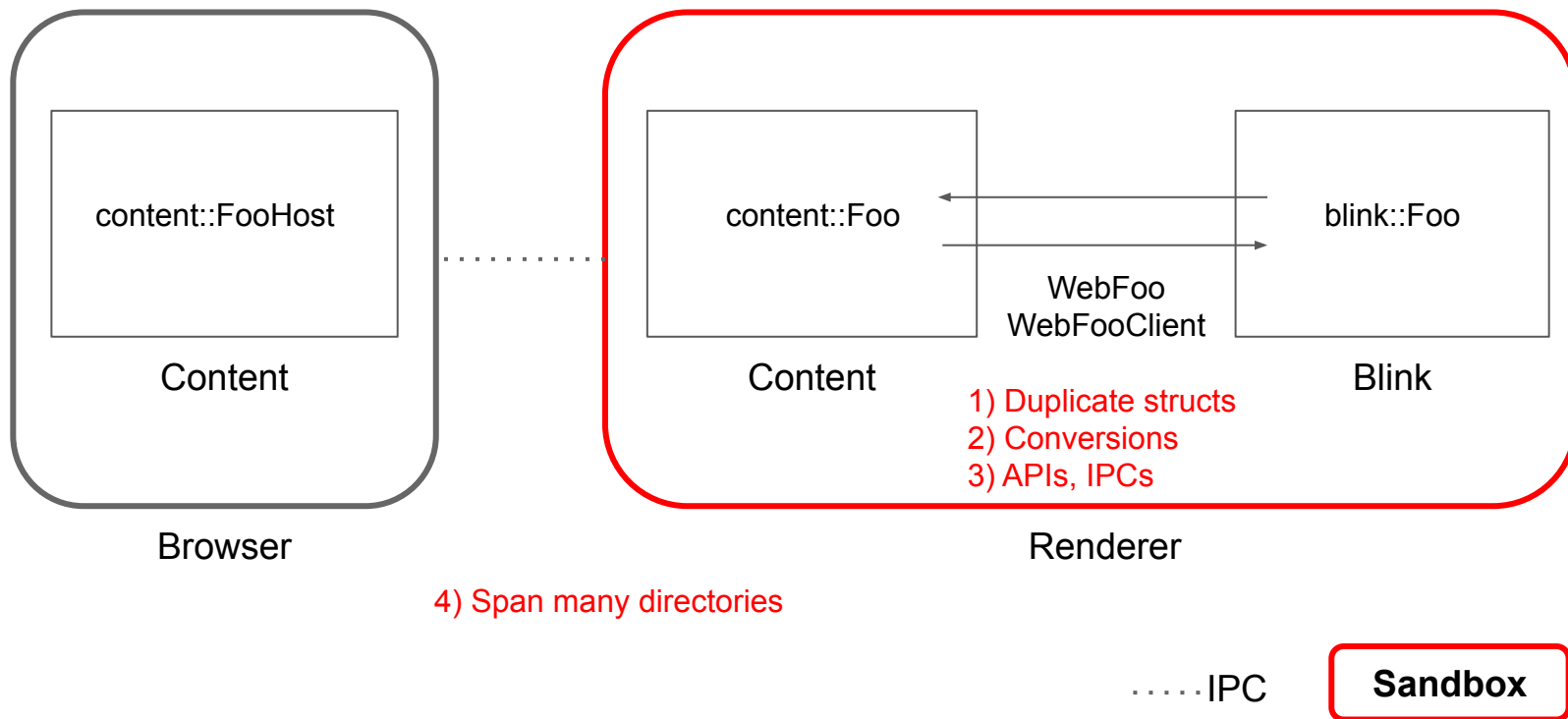2) Conversions

Renderer

· · · · IPC   **Sandbox**
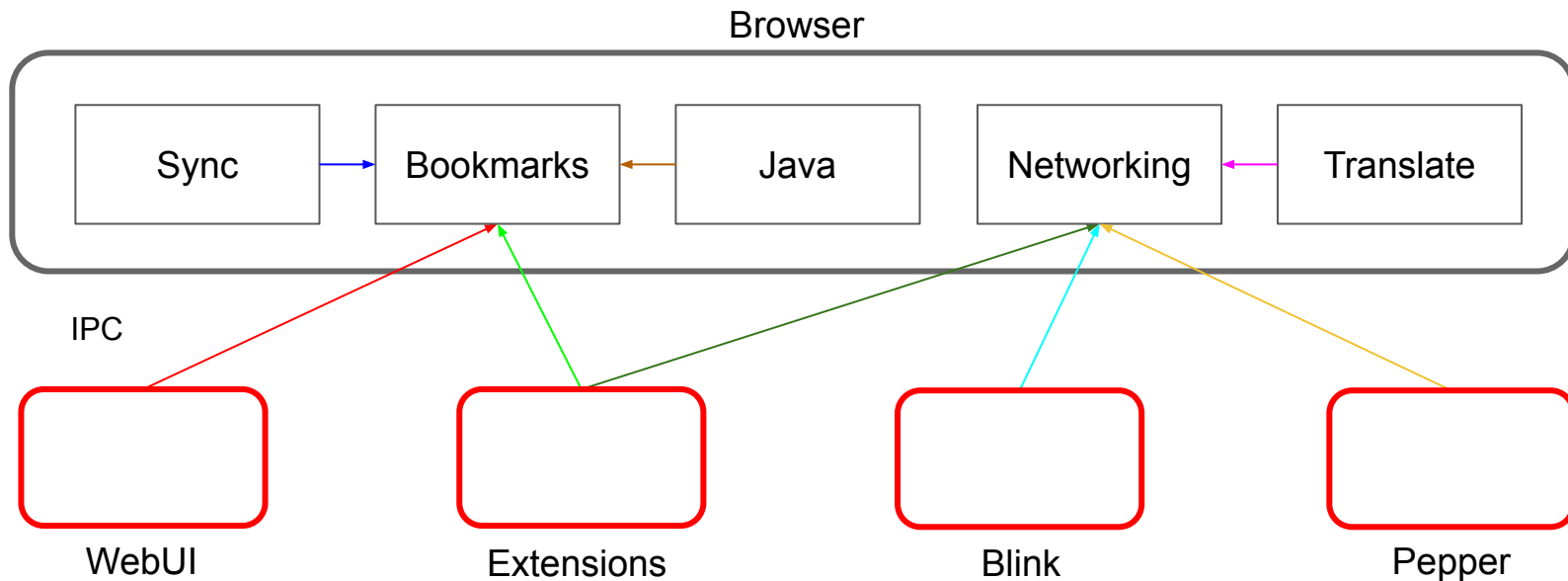
Motivation: Overhead
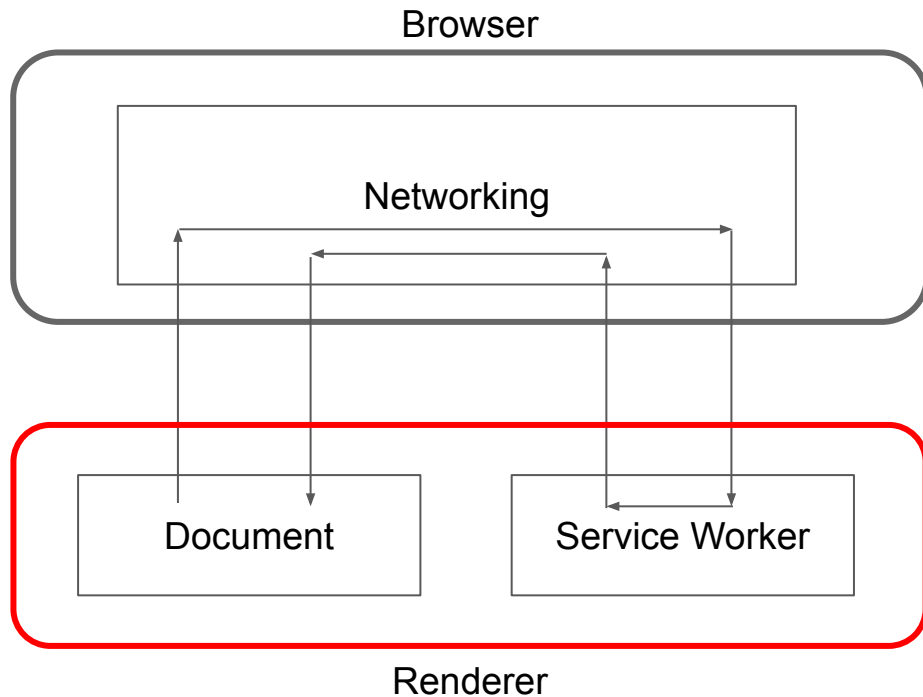
Motivation: Overhead

Motivation: Duplication

Motivation: Performance

Motivation: Performance
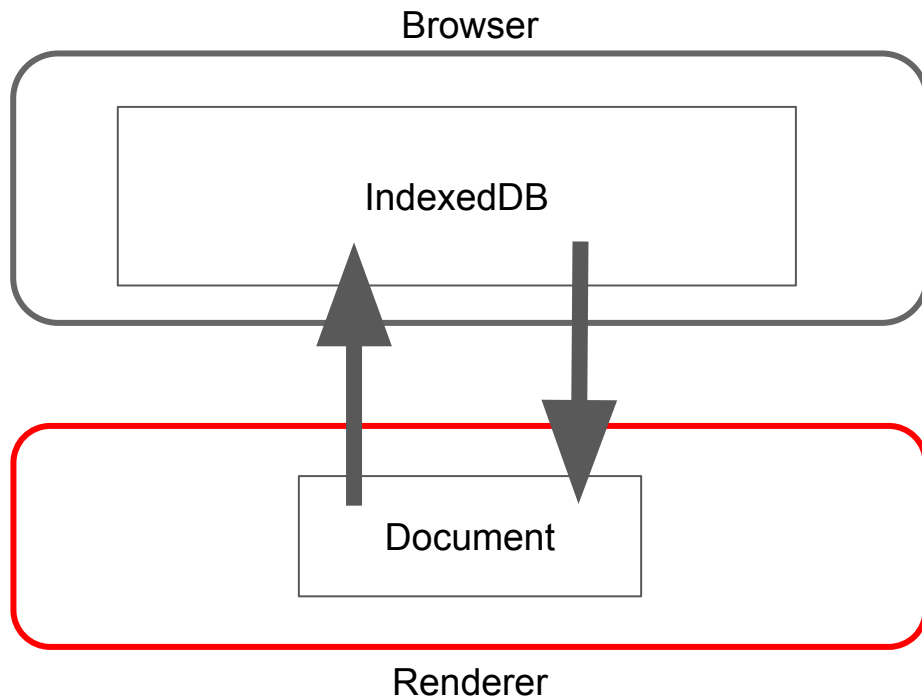
Motivation: Security
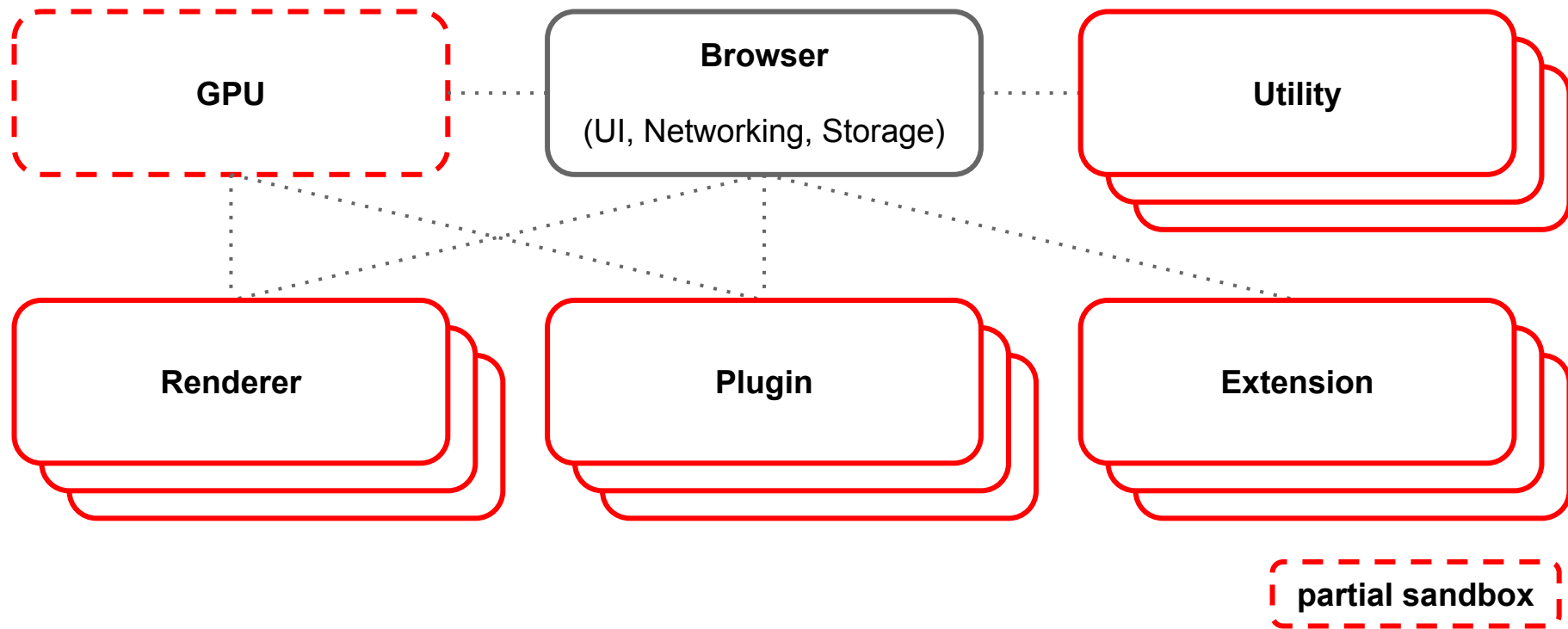
Goals

- speed
- security
- stability
- simplicity

How

- Mojofication: make IPCs directly from Blink
  - why couldn't we use old IPC system?
- Onion Soup
  - `content/renderer` can then move to `blink/`
  - web platform code in `content/child` is historical from when web workers ran in a separate process
- Servicification
  - `content/browser/{subsystems}` to `services/`
  - `content/browser/{web platform}` can move to `blink/`

Servicification

- microservice architecture
- well-defined
- reusable
- decoupled
- layered system
- exposed to Blink

Servicification: Projects

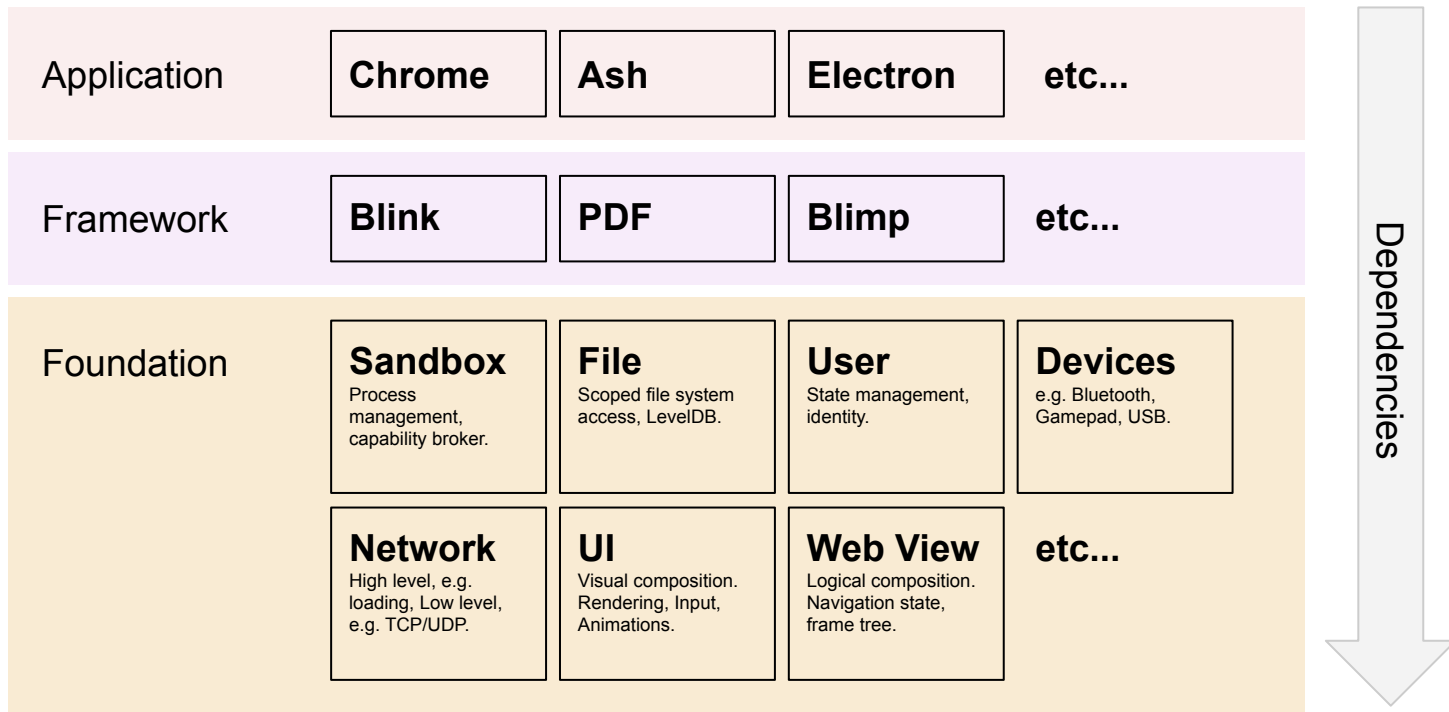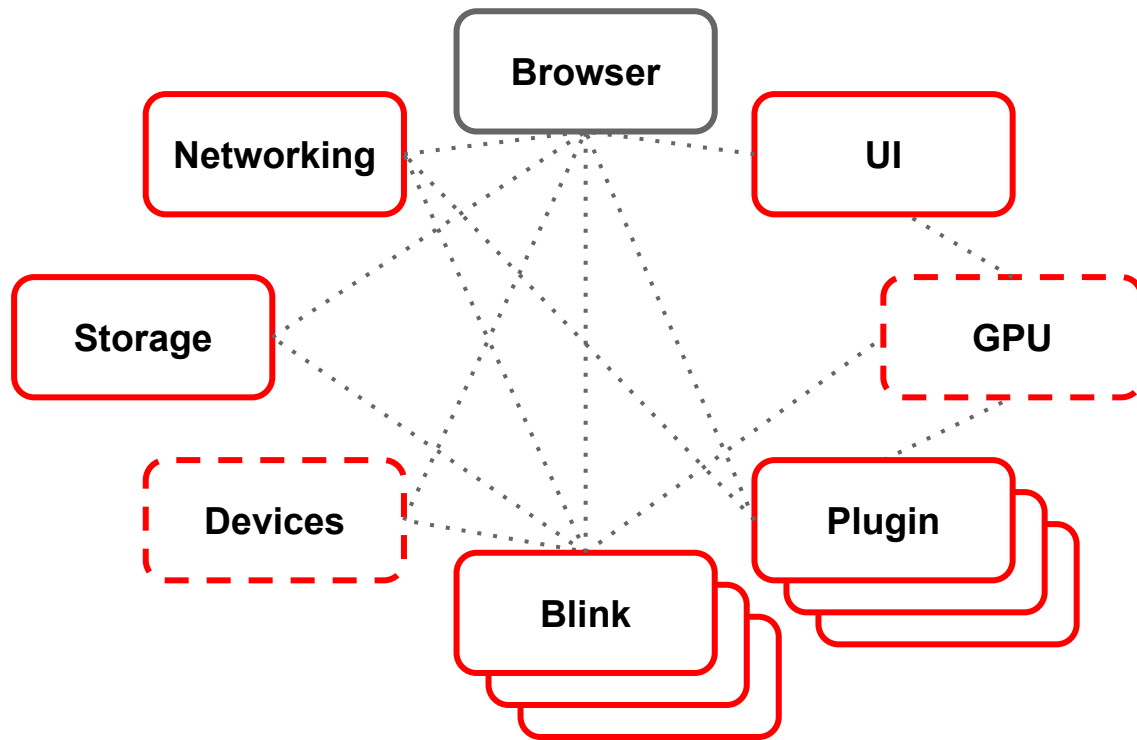| Completed	In Progress	Planned
File	Networking	Navigation
LevelDB	Download	Notifications
Preferences	UI/Viz	Sync
Device	ChromeOS Shell (Ash)	
Video Capture	Sandboxing	
	Permissions	
	Identity	
	Audio	

End State: Layer Cake

Application	**Chrome**	**Ash**	**Electron**	etc...

Framework	**Blink**	**PDF**	**Blimp**	etc...

Foundation

Sandbox Process management, capability broker.	**File** Scoped file system access, LevelDB.	**User** State management, identity.	**Devices** e.g. Bluetooth, Gamepad, USB.
Network High level, e.g. loading, Low level, e.g. TCP/UDP.	**UI** Visual composition. Rendering, Input, Animations.	**Web View** Logical composition. Navigation state, frame tree.	etc...

Dependencies ↓
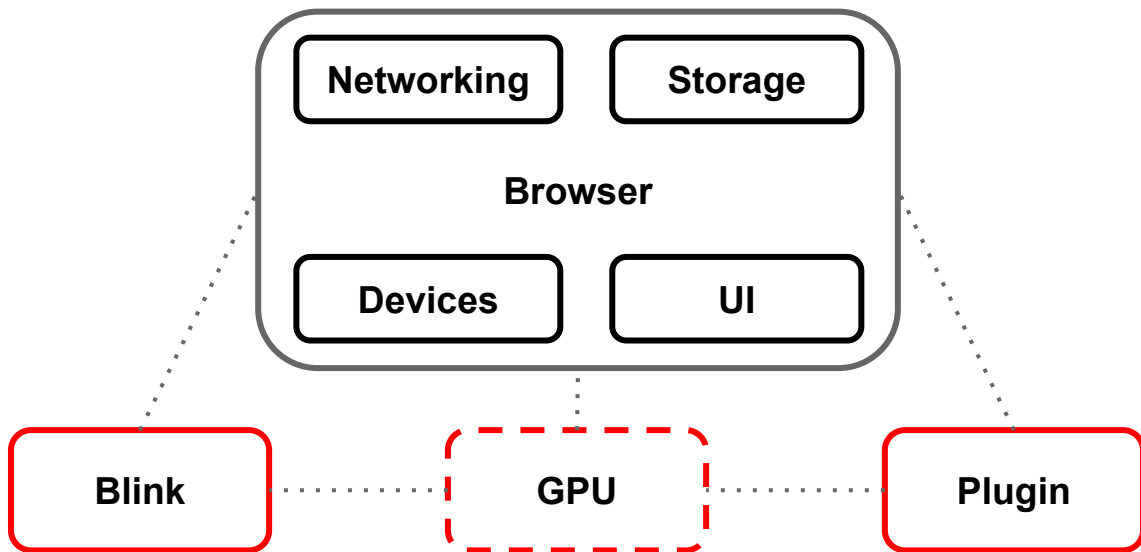
End State: Process Models

End State: Process Models

Deep Dive: LocalStorage

Current

Browser Process

content/browser/dom_storage

Renderer Process

content/renderer/dom_storage

third_party/WebKit/Source/modules/storage

Deep Dive: LocalStorage

After Onion Soup

Browser Process

Renderer Process

content/browser/dom_storage

third_party/WebKit/Source/modules/storage

Deep Dive: LocalStorage

After Onion Soup & Servicification

Renderer Process

```
third_party/WebKit/Source/modules/storage
```

```
LevelDB Service
```

Profile Process

Deep Dive: IndexedDB

Current

Browser Process

```
content/browser/indexed_db
```

Renderer Process

```
content/renderer/indexed_db

third_party/WebKit/Source/modules/indexeddb
```

IndexedDB

After Onion Soup

Browser Process   **Renderer Process**

content/browser/indexed_db

third_party/WebKit/Source/modules/indexeddb

IndexedDB

After Onion Soup & Servicification

Renderer Process

LevelDB Service

third_party/WebKit/Source/modules/indexeddb

End of Content?

- confident Content will be slimmed down substantially when we're done
- will it completely go away?
  - unclear at this point what will remain when we're done
  - if it's gone, great!
  - if not, we will have a much smaller module and we can study further steps to reduce or eliminate it

Questions