Microsoft Edge: Adopting and contributing to Chromium

Christian Fortini
Distinguished Engineer, Microsoft Edge
Microsoft Edge on Chromium: Why?

- Better web compatibility for our customers
- Less fragmentation of the web for web developers
- Deliver and update for all supported versions of Windows and on a more frequent cadence
- Join forces to evolve the web
Our Open Source Principles

1. We are making this decision for the long term

2. When seeking improvements in the web platform, our default position will be to contribute

3. We will respect the architectural requirements and engineering approaches used in Chromium

4. We believe the evolution of the web is best served through standards bodies
# Areas of Focus

<table>
<thead>
<tr>
<th>Accessibility</th>
<th>Editing</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM64</td>
<td>Fonts</td>
<td>Tooling</td>
</tr>
<tr>
<td>Authentication</td>
<td>Layout</td>
<td>Touch</td>
</tr>
<tr>
<td>Battery life</td>
<td>Scrolling</td>
<td>Web Standards</td>
</tr>
</tbody>
</table>
Building Edge on Chromium
### Engineering

<table>
<thead>
<tr>
<th>Goma</th>
<th>Homegrown ES</th>
<th>LUCI</th>
</tr>
</thead>
</table>
| - Created a Goma service on Azure | - Handles PRs from our team  
- Merges ~450 changes from upstream Chromium daily  
- Resolves ~75 conflicts daily  
- Rolling builds  
- Daily release builds  
- Based on Jenkins & Azure | - Exploring porting LUCI to Azure  
- Looking at layers of abstraction: GAE, NDB  
- Should scale well to our needs  
- If successful, we are interested in contributing to LUCI |
• **Telemetry**
  UMA/UKM uploaded to our instrumentation services (1DS)

• **Crash reporting**
  Connected to our Watson service

• **Experimentation**
  Integrated with our Experimentation and Configuration Service (ECS)
We upload UMAs (UKMs soon) using 1DS client SDK
Takes care of upload and retries, populating common schema fields, privacy requirements

Privacy requirements include
- Privacy classification: critical, measure, telemetry
- Transparency requirements (GDPR)
**New Edge Reporting Service**

- Work from persisted log
- Deserialize using protobuf_full
- Split to 1DS schema

**Chromium Reporting Service**

- Chromium Uploader
  - Google

**Edge Reporting Service**

- Persisted Log
  - Compress Rotate
  - Split
- Decompress Deserialize Parse
- Split

**1DS Client**

- Microsoft
Potential Improvements:

- Work directly from in-memory log
- Move off main thread
- Use differential privacy reporting (RAPPOR)
Contribution Areas
Contributions

In-progress:
- Accessibility (UIA)
- High contrast
- Caption Styling (Web VTT)
- Native caret browsing
- ARM 64
- TSF1

Other areas we would like to help with:
- PDF enhancements
- Battery life
- Smooth Scrolling
- Editing
- Layout
- Dev tools
- Web Authentication

~300 merges so far
 Implements:

- Auto-correction
- Keyboard suggestions
- Shape writing
- IME reconversion after composition

Built on existing (inactive) implementation
Available on Win7, 8.1, 10
Areas we would like to contribute

• Fundamentals: bug fixes, test coverage
• Accessibility enhancements: better screen reader, high contrast
• Smooth Scrolling: HTML based layout
• Simplification of process model
Investigation: Power Reduction

3 areas of investigation:
• High resolution timer in message pump
• Disk cache during video playback
• Audio offloading to hardware
Memory Reduction in DLL Prefetching

On Windows 10 Insider:
- Rely on OS (size caps increased)

On Win8 – Win10 1809:
- Chromium pre-reads file in MapFile memory -> forces a copy
- Instead map file as Image, and fetch using PrefetchVirtualMemory
- Savings for chrome.dll: 80+ MB
Investigation: Smooth Scrolling

Scrollbar latency
- Intent to Implement
  impl-threaded scrollbars

Missed frames (investigating)
- Composition thread priority
- GPU contention, scheduling
  misses, IPC delays
Other Potential Contributions

**Smooth Scrolling**
We would love to help with:
- BlinkGen Property Tree (BGPT)
- Composition After Paint (CAP)
- Independent rendering

**LayoutNG**
We went through a similar transition in EdgeHTML in IE9
Other Edge Features
Edge Features

- PlayReady DRM
- Services integration
- Single Sign-On
Why PlayReady?

- 4k streaming DRM content (e.g., Netflix)
- Hardware decryption/decoding
- Software decryption in protected process (LPAC)
- Power savings: offloaded audio, video batching

Edge supports PlayReady and Widevine
PlayReady DRM

- Supported by Windows Media Foundation
- New MediaEngineRenderer and CDM in sandbox
- Uses existing support of Direct Composition (DCOMP)
PlayReady DRM

Overall, after ramping up on the architecture, it was relatively easy to plug-in PlayReady DRM

 Mostly small issues:
• Chromium kills renderer after 1 min of pause
• Sites get confused because Edge 1st to support PlayReady and Widevine
• Still fixing bugs!
Services we replaced or turned off

Safe browsing
Nearby messages
Link Doctor
Ad blocking
User data sync
Spellcheck
Suggest
Translate
SmartLock
Form Fill
Push Notifications
WebStore
Extension Store
Maps Geolocation
Google Now

Speech input
Google Pay
Drive API
Chrome OS hardware id
Device registration
Google Maps Time zone
Google Cloud Storage
Cloud Print
Google DNS
Supervised Profiles
Address Format
Network Location
Network Time
Favicon service
Google Cloud Messaging

Single sign-on (Gaia)
Content Hash Fetcher
Flighting Service
Component Updater Service
RAPPORT service
Chrome OS monitor
calibration
Chrome OS device
management
Android app password sync
Offline Page Service
Feedback
Domain Reliability Monitoring
Data Reduction Proxy
Chrome Cleanup

Developer Tools Remote Debugging
iOS Promotion Service
One Google Bar Download
Brand Code Configuration Fetcher
WebRTC Logging
Captive Portal Service
Edge supports:
- MSA or AAD identities
- App to browser SSO
- Browser to app SSO
- OS to browser SSO
- Browser to site SSO

Identity and Single Sign-On (SSO)

Sign-in Menu ➔ SigninViewController ::ShowSignIn ➔ EdgeAuthenticationManager ::StartSignIn ➔ EdgeAuthenticationManager ::StartSignIn callback ➔ Populate User Info into UI

Sign-in Button ➔ EdgeAuthenticationManager ::StartSignIn ➔ OneAuth Login UX
- Token Fetching
- App and OS SSO
- Website SSO
- ...

OAuth2TokenServiceDelegate ::CreateAccessTokenFetcher ➔ EdgeAccessTokenFetcherImpl

Async

AccessToken Needed for Service Call ➔ EdgeAccessTokenFetcherImpl HandleAccessTokenFetch callback ➔ Proceed with Service Call

Edge specific
Thank you for a great partnership!
We made hundreds of changes to Chromium to produce Edge
Contributed 300+ merges so far and planning on contributing much more!
We’re building expertise in many areas of the code base
Overall, building Edge on Chromium was a relatively smooth process
Looking forward to contributing more!
Thank you!