Microsoft

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

- Accessibility
- ARM64
- Authentication
- Battery life
- Editing
- Fonts
- Layout
- Scrolling
- Security
- Tooling
- Touch
- Web Standards

Building Edge on Chromium

Engineering

Goma

- Created a Goma service on Azure
- Utilizing Azure load balancing

Homegrown ES

- 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

LUCI

- 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

Instrumentation

- **Telemetry**

  UMA/UKM uploaded to our instrumentation services (1DS)

- **Crash reporting**

  Connected to our Watson service

- **Experimentation**

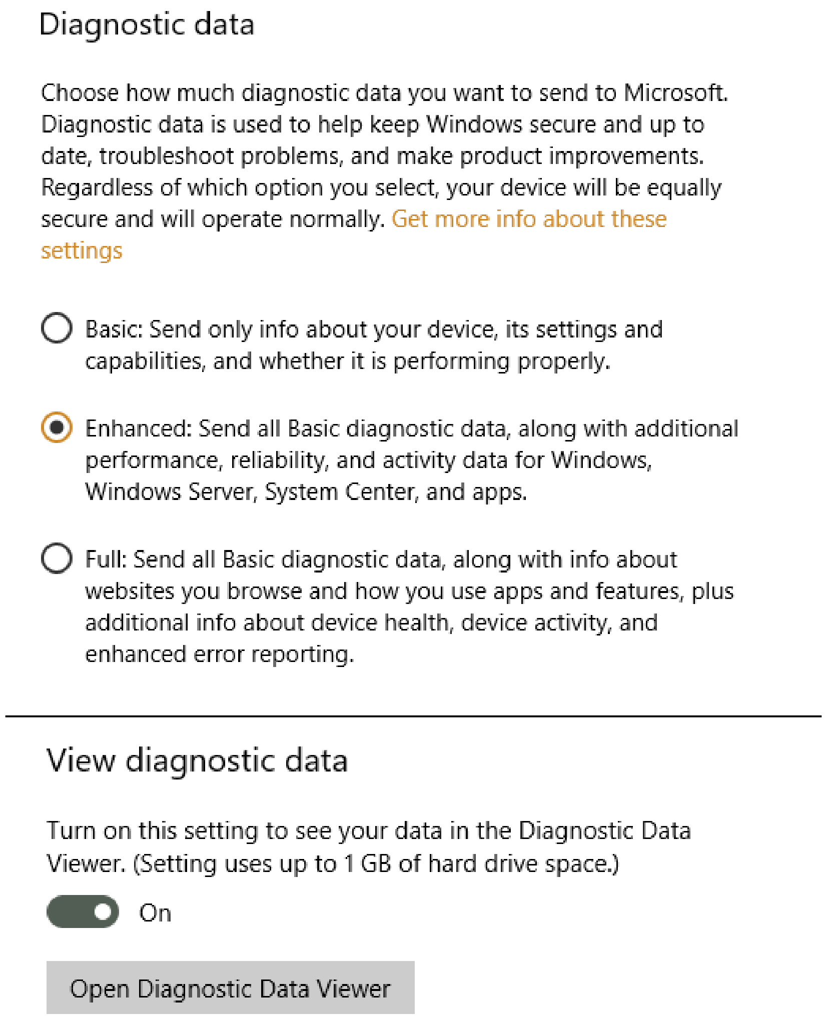
  Integrated with our Experimentation and Configuration Service (ECS)

UMA/UKM

- **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)



Diagnostic data

Choose how much diagnostic data you want to send to Microsoft. Diagnostic data is used to help keep Windows secure and up to date, troubleshoot problems, and make product improvements. Regardless of which option you select, your device will be equally secure and will operate normally. Get more info about these settings

○ Basic: Send only info about your device, its settings and capabilities, and whether it is performing properly.

● Enhanced: Send all Basic diagnostic data, along with additional performance, reliability, and activity data for Windows, Windows Server, System Center, and apps.

○ Full: Send all Basic diagnostic data, along with info about websites you browse and how you use apps and features, plus additional info about device health, device activity, and enhanced error reporting.

View diagnostic data

Turn on this setting to see your data in the Diagnostic Data Viewer. (Setting uses up to 1 GB of hard drive space.)

On

Microsoft

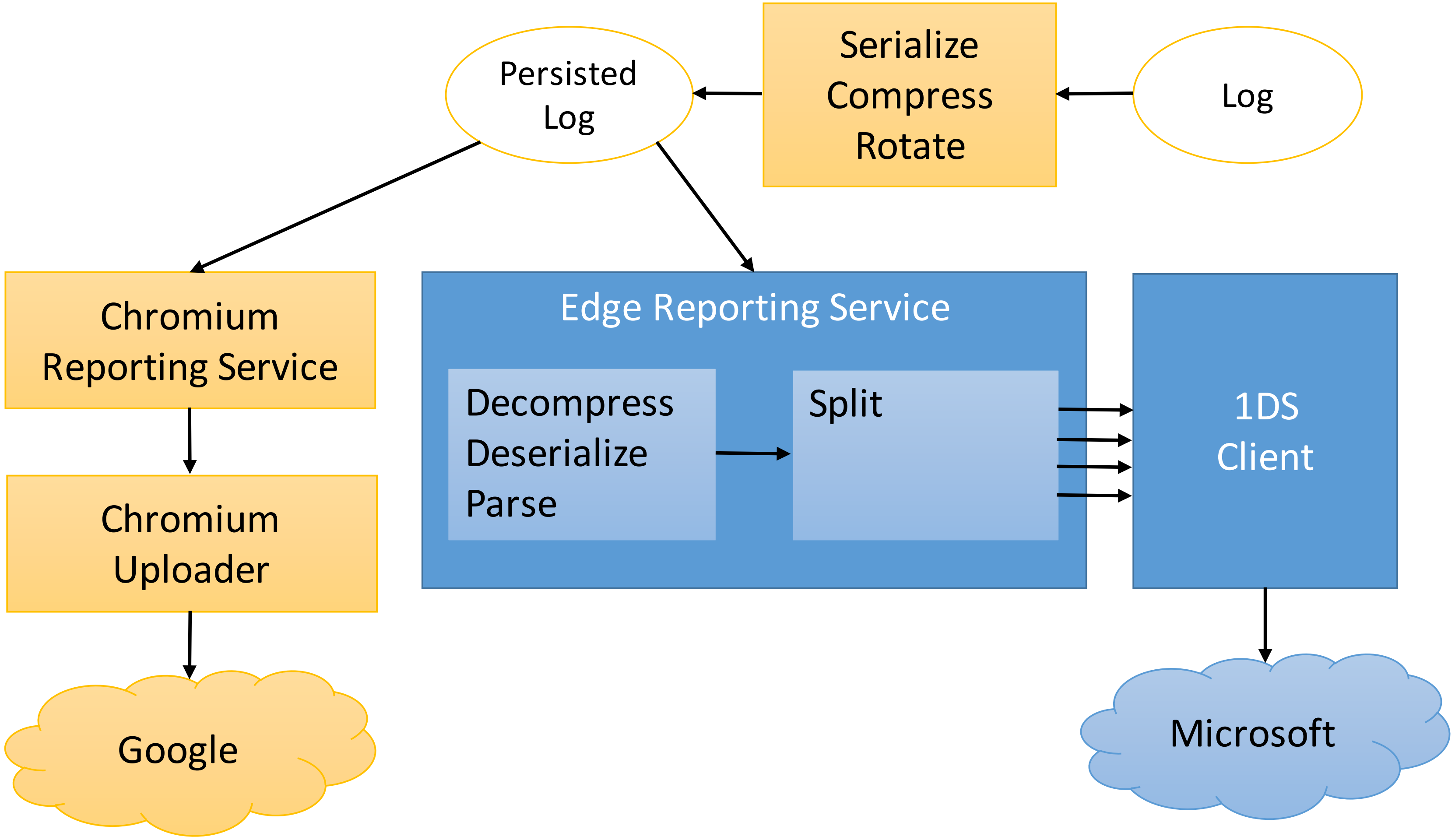
UMA/UKM

New Edge Reporting Service

Work from persisted log

Deserialize using protobuf_full

Split to 1DS schema



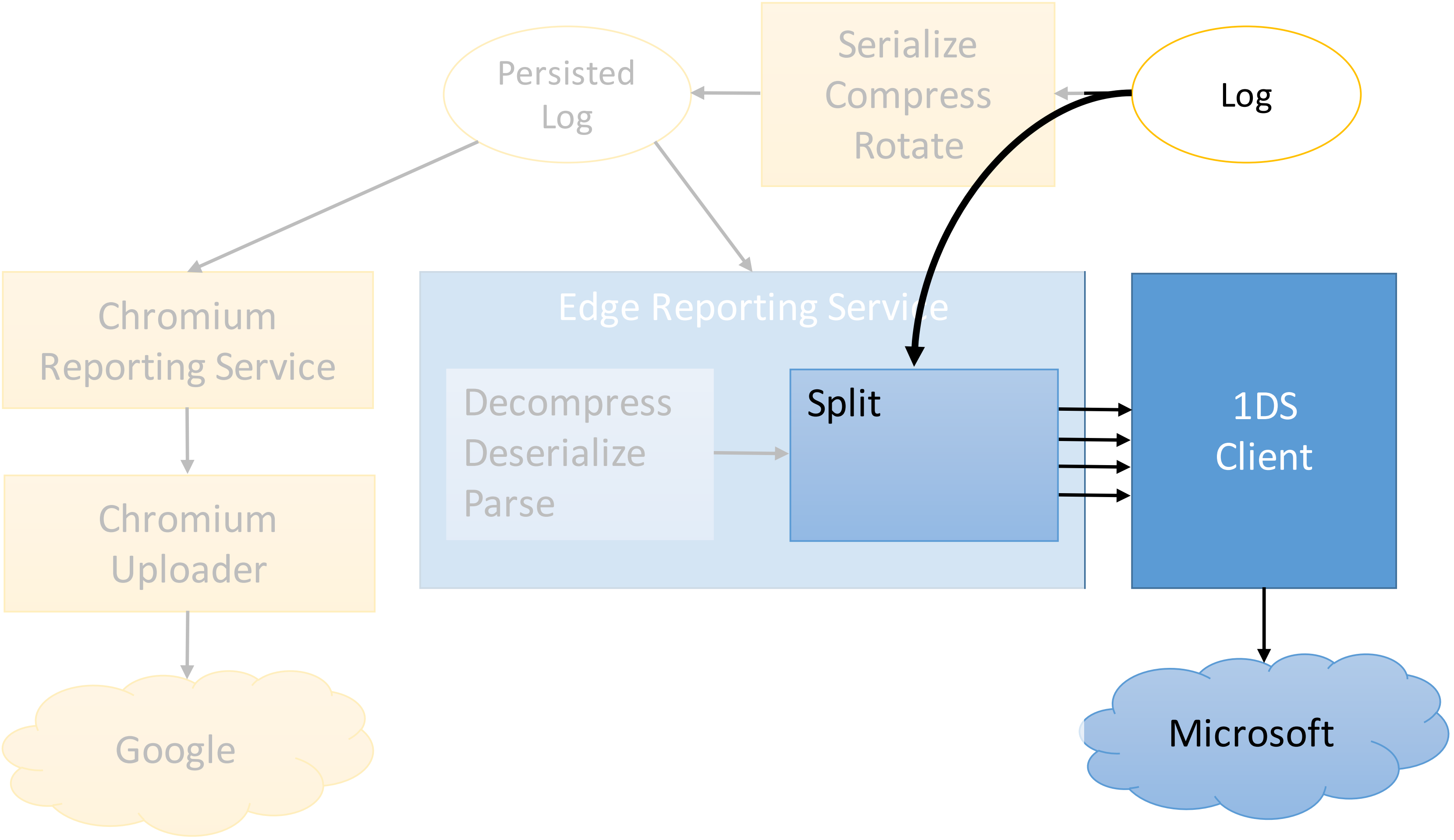
UMA/UKM

Potential Improvements:

Work directly from in-memory log

Move off main thread

Use differential privacy reporting (RAPPOR)?



Contribution Areas

Contributions

~300 merges so far

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

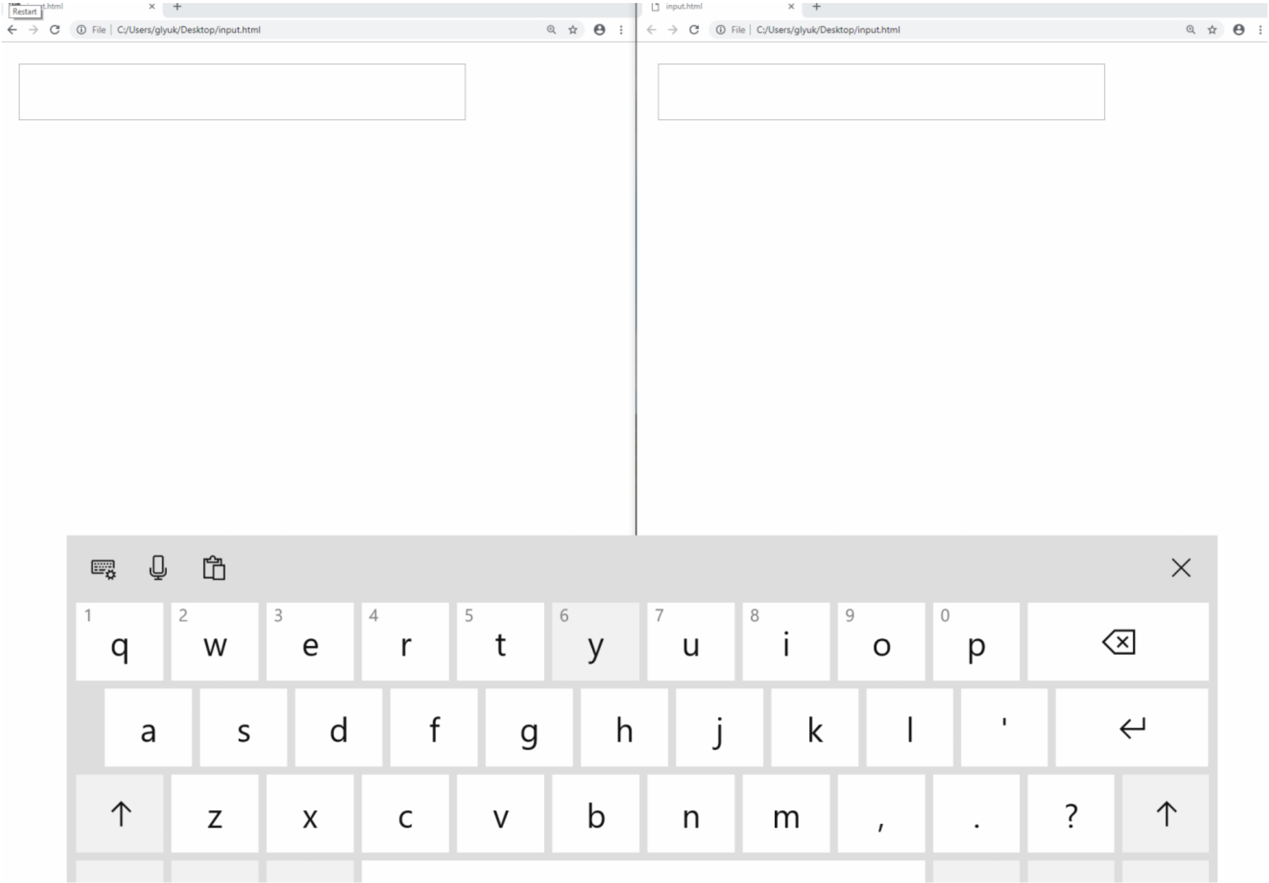
TSF1

Implements:

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

Built on existing (inactive) implementation

Available on Win7, 8.1, 10


Restart
input.html
File | C:/Users/glyuk/Desktop/input.html
input.html
File | C:/Users/glyuk/Desktop/input.html
1 q 2 w 3 e 4 r 5 t 6 y 7 u 8 i 9 o 0 p
a s d f g h j k l '
z x c v b n m , . ?

PDF Enhancements

Areas we would like to contribute

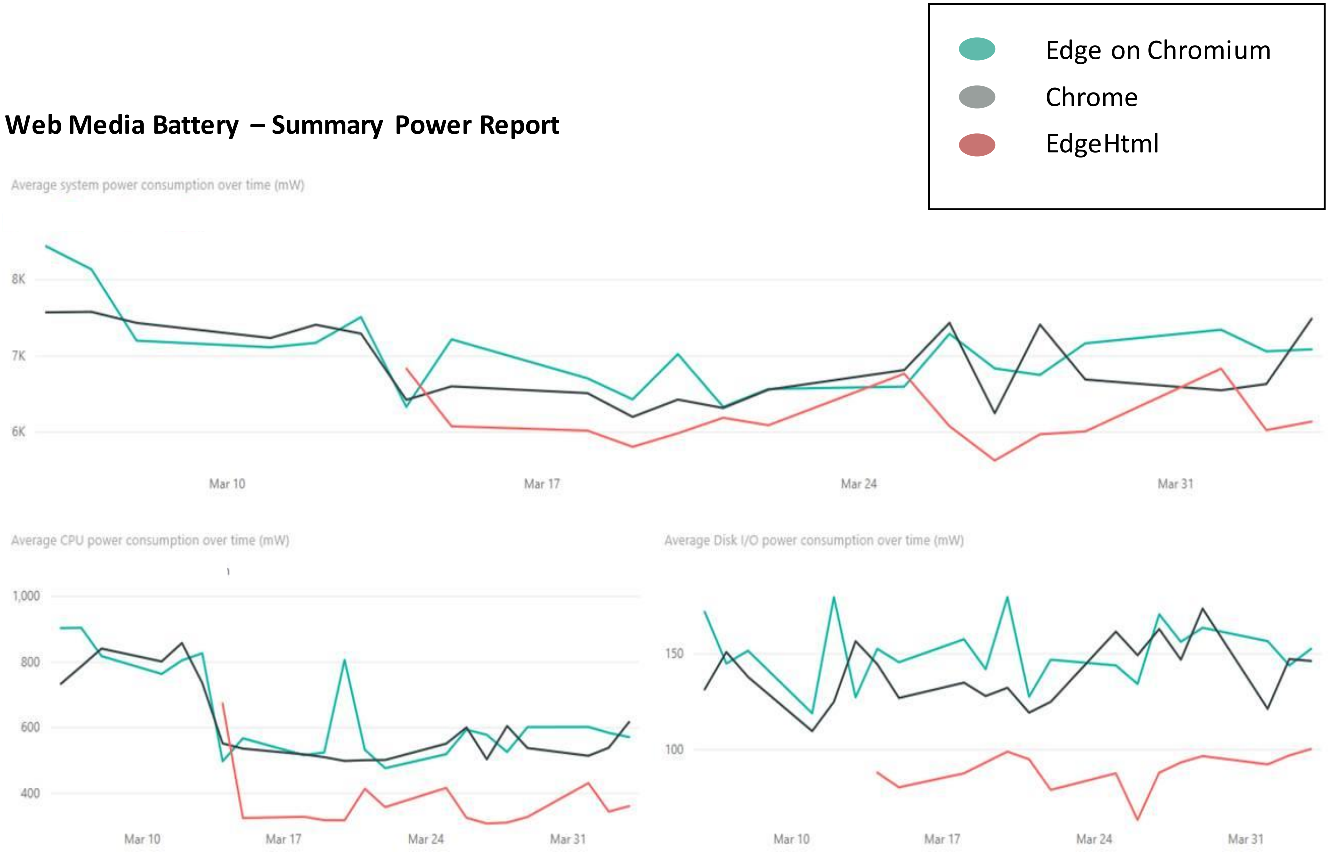
- Fundamentals: bug fixes, test coverage
- Accessibility enhancements: better screen reader, high contrast
- Smooth Scrolling: HTML based layout
- Simplification of process model

Investigation: Power Reduction

3 areas of investigation:

- High resolution timer in message pump
- Disk cache during video playback
- Audio offloading to hardware

Web Media Battery – Summary Power Report



Memory Reduction in DLL Prefetching

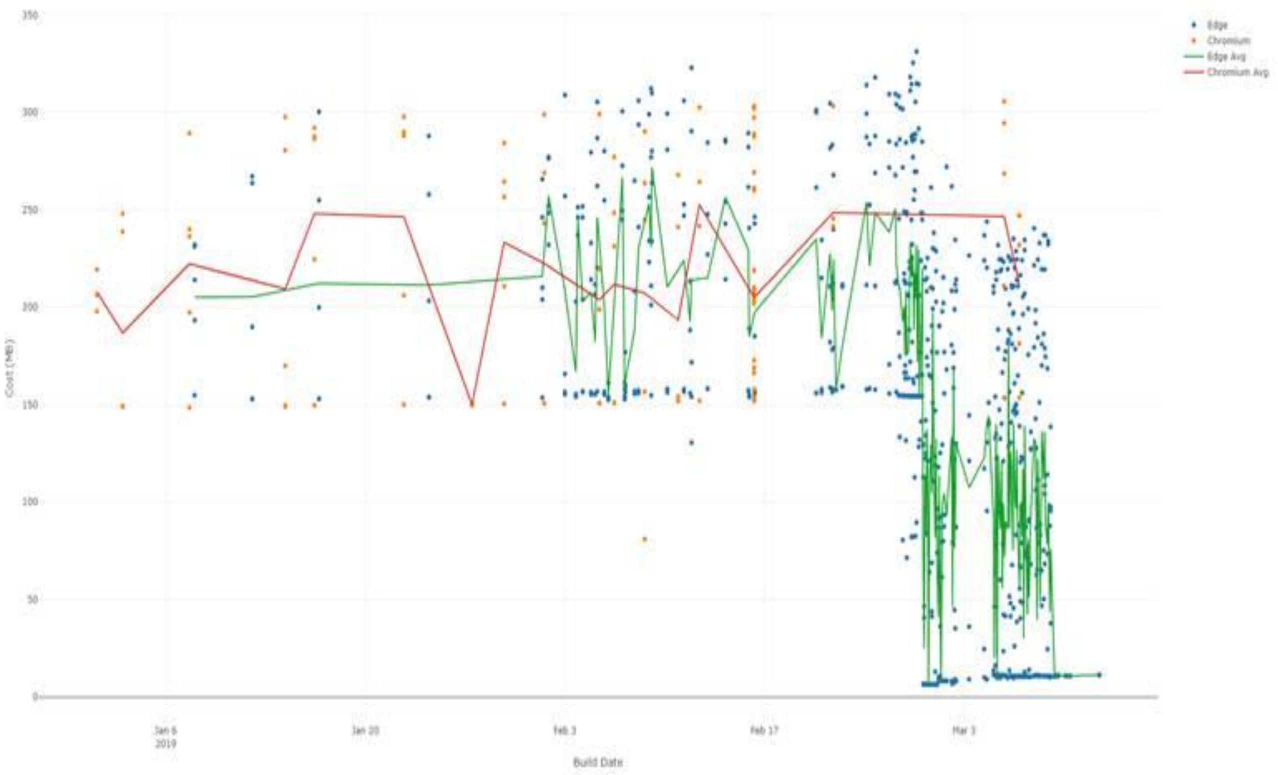
On Windows 10 Insider:

- 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

MapFile System Cost



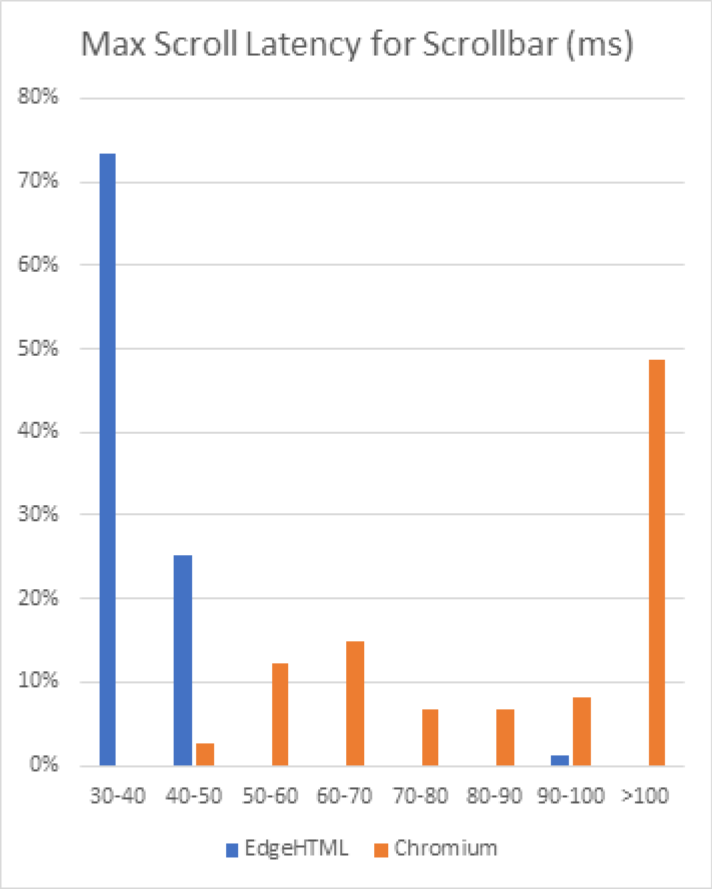
Investigation: Smooth Scrolling

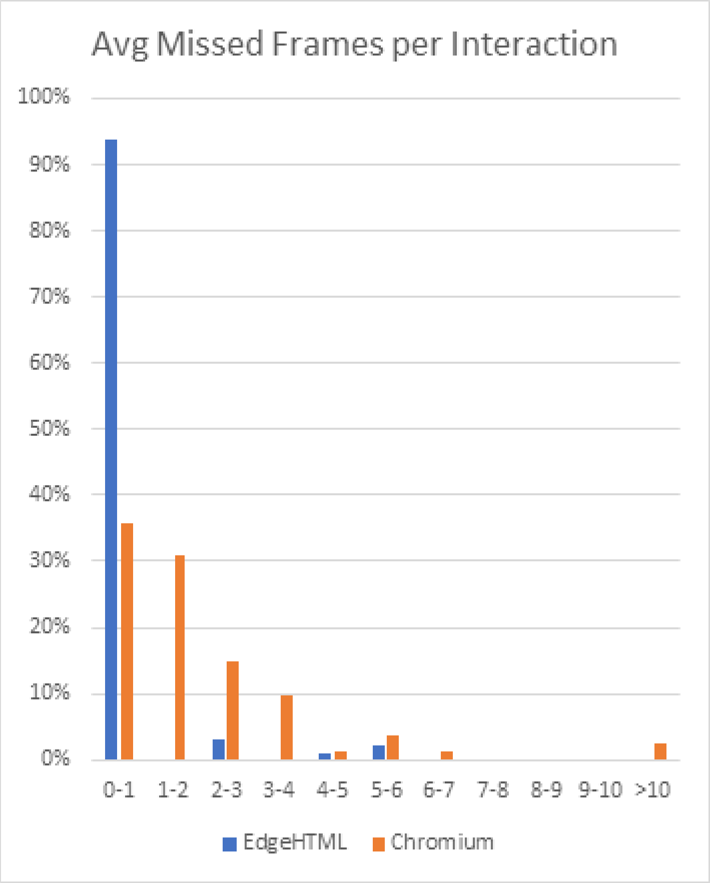
Scrollbar latency

- Intent to Implement impl-threaded scrollbars

Missed frames (investigating)

- Composition thread priority
- GPU contention, scheduling misses, IPC delays





Microsoft

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

PlayReady DRM

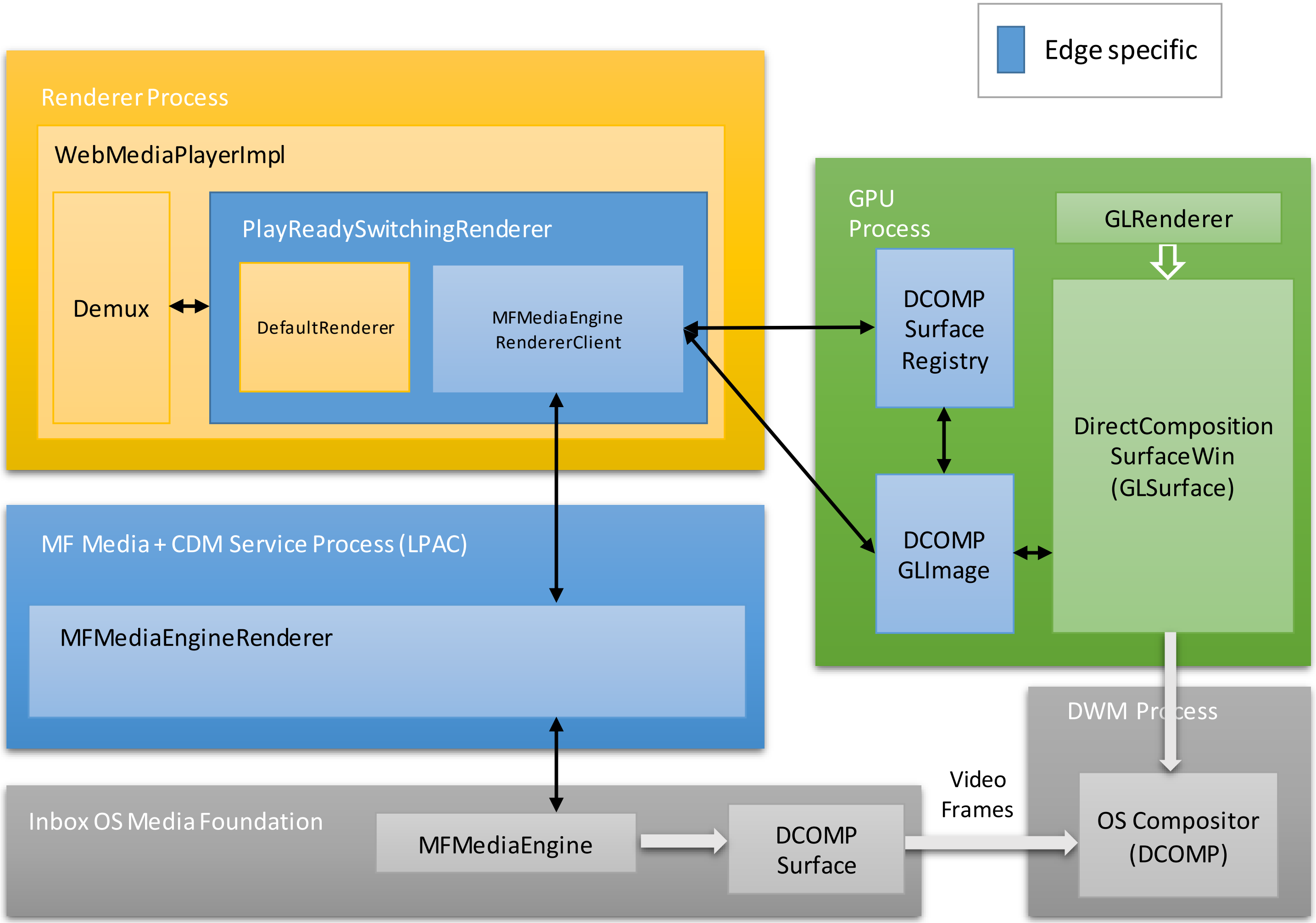
Why PlayReady?

- 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

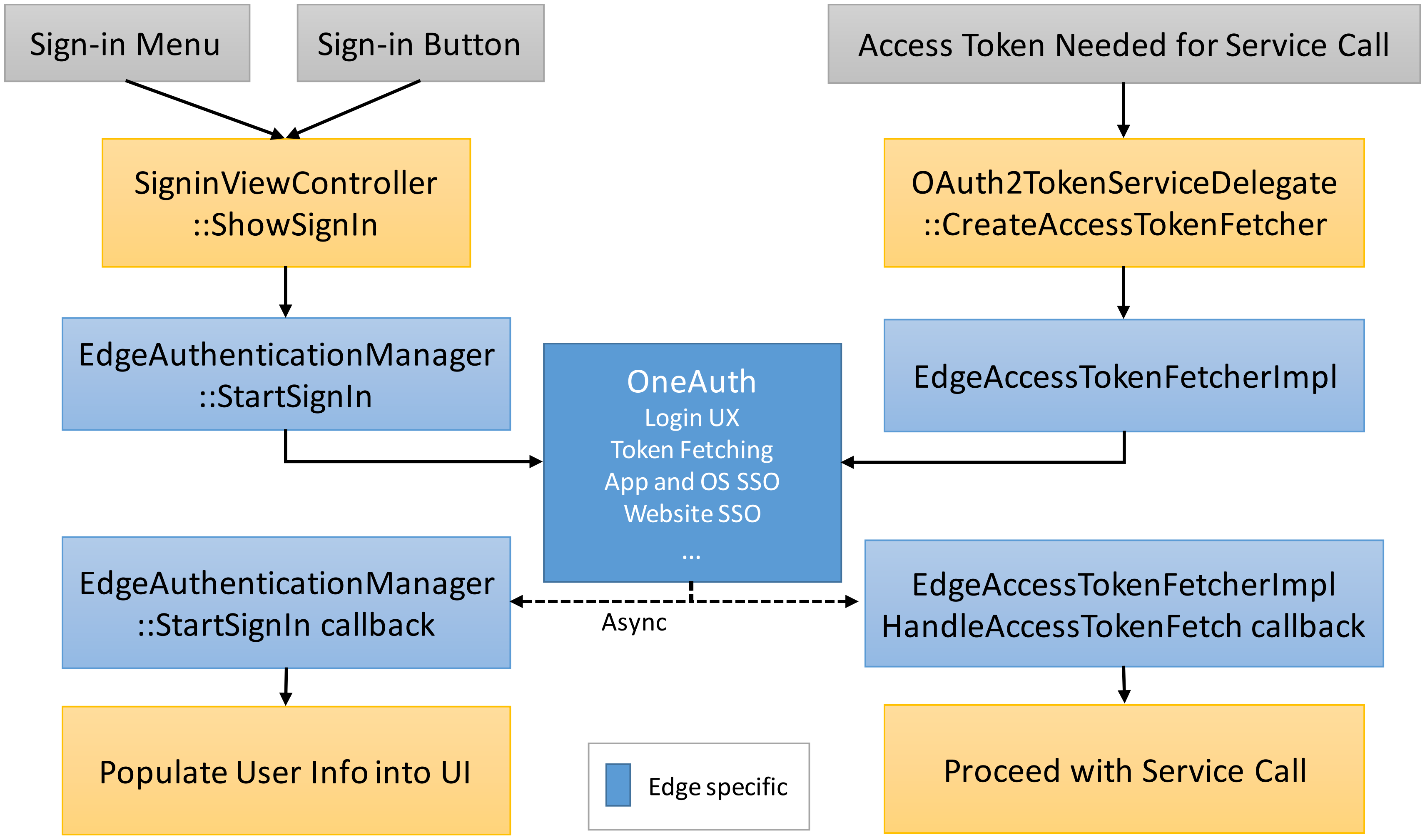
Single sign-on (Gaia)
Content Hash Fetcher
Flighting Service
Component Updater Service
RAPPORT service
Chrome OS monitor calibration
Chrome OS device management
Android app password sync
Offline Page Service
Feedback
Domain Reliability Monitoring
Data Reduction Proxy
Chrome Cleanup

Developer Tools Remote Debugging
iOS Promotion Service
One Google Bar Download
Brand Code Configuration Fetcher
WebRTC Logging
Captive Portal Service

Identity and Single Sign-On (SSO)

Edge supports:

- MSA or AAD identities
- App to browser SSO
- Browser to app SSO
- OS to browser SSO
- Browser to site SSO


Sign-in Menu
Sign-in Button
Access Token Needed for Service Call
SigninViewController ::ShowSignIn
OAuth2TokenServiceDelegate ::CreateAccessTokenFetcher
EdgeAuthenticationManager ::StartSignIn
OneAuth
Login UX
Token Fetching
App and OS SSO
Website SSO
…
EdgeAccessTokenFetcherImpl
EdgeAuthenticationManager ::StartSignIn callback
Async
EdgeAccessTokenFetcherImpl HandleAccessTokenFetch callback
Populate User Info into UI
🟦 Edge specific
Proceed with Service Call

Recap

- 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!

Microsoft