WAM: an embedded web runtime history

Introduction and retrospective

February 4th, 2023. FOSDEM @ Brussels

José Dapena Paz <jdapena@igalia.com>
About me (1/3)

José Dapena Paz “dape”

Software engineer at Igalia for 21 years, and one of its founders.
About me (2/3)

From Galicia, Spain.
About me (3/3)

Member of Igalia Chromium team.

Some projects:

- LG webOS web engine integration (2012-today)
- WAM for AGL (2018-2022)
- Nokia Meego and Maemo (2005-2011)
Igalia

• Open source consultancy, focused on the web ecosystem.
• +100 engineers all around the world.
• 2nd main contributor to Chrome, right after Google.
• 2nd main contributor to WebKit, right after Apple.
• Teams around web engines and platform, graphics, compilers, ...
Why this talk?

I started working on webOS in October 2012 (10 years ago!).

10 years feels like a good time to write about history.
Agenda

- What is WAM?
- Where is it used?
- How it works?
- What is good about WAM?
- Retrospective
- Future
What is WAM
What is WAM

WAM = Web Application Manager
What is WAM

WAM = Web Application Manager

• LG webOS web runtime.
What is WAM

WAM = Web Application Manager

- LG webOS web runtime.
- On top of Chromium.
What is WAM

webOS

An Operating System for embedded products.

It is web centric: web applications are first class citizens.
What is WAM

webOS

Yocto / Openembedded Core.

Wayland / QML / Maliit.

Unified media server.

Luna bus for IPC.
What is WAM

WAM = Web Application Manager

- LG webOS web runtime.
- On top of Chromium.
What is WAM

**WAM** = **Web Application Manager**

- LG webOS web runtime.
- On top of Chromium.

WAM is the center piece of web experience in webOS.
Where is it used?
Where WAS it used?

- HP Touchpad
- Some Palm phones
Where is it used?

LG webOS TV, from 2013.
Where is it used?

Some LG products and experiments that use or used webOS and WAM:
Where is it used?

Where is it used?

Linux Foundation Automotive Grade Linux (AGL) web runtime.

Port done adapting to the AGL application framework.
How it works?
How it works?
What's good about WAM?
What's good about WAM?

Support of web platform standards

**WAM** is based on Chromium: high support for web platform standards.

- Document model, JS, CSS.
- Device APIs.
- Graphics APIs.
- Media, WebRTC
- Storage, security
What's good about WAM?

Application lifecycle

**WAM** runtime manages the running web applications:

- Launching, closing
- Suspend/resume
- Recovery
What's good about WAM?

Performance

A goal of WAM is using the system resources efficiently:

- CPU and GPU memory.
- CPU usage.
- Launch time optimization.
What's good about WAM?

Security

Local packaged applications and remote web applications.

Following web security origin model.

Application manifest permissions.
What's good about WAM?

Developer tools

- Web inspector and developer tools
- Chromium tracer
What's good about WAM?

Proven technology

Shipped in millions of LG smart TVs...

... and +10 years of experience.

Nowadays also adopted for the AGL reference platform.
What's good about WAM?

Open source license

https://webosose.org

https://github.com/webosose/wam

Apache 2.0 license
Retrospective
Retrospective

Caveat

I have been working on webOS web integration since October 2012.

History of WAM seems to start around 2008.

I will mostly focus on what I lived.
Retrospective

From HP/Palm to LG TV

2010: HP bought Palm...

2011: ... Then HP decides to cancel new webOS products. Last device: HP Touchpad.
Retrospective

From HP/Palm to LG TV (2/2)

2012: partnership LG+HP for using webOS on LG SmartTV.

2013: LG acquires Palm software development unit from HP: LG Silicon Valley Labs is born.

2014: LG webOS based TV are introduced in CES Las Vegas. TV are released a few months later.
Retrospective

webOS and Open Source: Open WebOS (1/2)

June 2012: HP/Palm releases OpenWebOS

Code still available at https://github.com/openwebos

2012: after LG acquisition, OpenWebOS is not maintained anymore!
So:

- I guess it was useful to attract LG to invest in webOS.
- LG preferred to focus on making LG webOS TV happen.

My view: the webOS TV effort was a huge challenge. No time for anything else.
Retrospective

Open Source: webOS OSE (1/3)

June 2018: LG releases webOS Open Source Edition

See https://webosose.org

Focus:

- Allow prototyping ideas and products.
- For students, independent developers.
- Create a community around it.
Retrospective

Open Source: webOS OSE (2/3)

It is active nowadays.

Hardware target: Raspberry PI (nowadays PI 4 models).

My view:

- It simplified testing new ideas.
- Integration of parts for AGL for automotive, ROS for robotics.
Retrospective

Open Source: webOS OSE (3/3)

Do you want to integrate web UI in your experimental product? webOS OSE can help!
Retrospective

From webOS to AGL (1/2)

Collaboration LG + Igalia + Linux Foundation.

Before 2018: Igalia assisted Chrome browser work with Wayland

May 2018: LG Silicon Valley Labs experiment to port WAM to AGL
Retrospective

From webOS to AGL (2/2)

The 2018 experiment proved it was possible.

Focus on a web-only demo:

- System UI + web applications
- Integration of third party applications.
- Integration with system services.

In last 4 years WAM evolved with AGL.
Retrospective

From QtWebKit to webOS WebView (1/2)

2012: WAM was based on QtWebKit

2012-2013: Moved to QtWebKit2 (multiprocess architecture).

2014-2015:

• Moved to Blink following market trend after Blink forked from WebKit.
• Based on QtWebEngine.
Retrospective

From QtWebKit to webOS WebView (2/2)

2015–today:

- Created webOS WebView layer and dropped QtWebEngine.
- Why?
  - No strong QWE specific use cases.
  - Qt started moving towards commercial and tighter open-source licensing policies.
Retrospective

From Qt to Qt-less (1/2)

WAM has been based on Qt for a long time.

But...

- GPLv3 licensing of Qt was problematic for some stakeholders (but not LG).
- C++, STL and Chromium C++ libraries grew better and better. Not much benefit on Qt.
Retrospective

From Qt to Qt-less (2/2)

From 2021, WAM does not depend on Qt anymore.

- Based on STL and other C++ libraries + Glib.
- QMake -> CMake.
Retrospective

Stability (1/2)

But... essentially, main ideas of WAM did not change in last 10 years.
Retrospective

Stability (2/2)

It proved useful, and architecture flexible enough to adapt to:

- Web engine changes.
- Dependencies changes.
- New products
- Even OS changes.
Future
Future

webOS is here to stay!

LG is increasing the bet for webOS on SmartTV through webOS Hub.
Future

Some topics...

- GCC or Clang?
- How to improve upgrade cycles?
Final remarks
Final remarks

• +10 years of project. More to come!
• Proved useful for many products and millions of users.
• A great piece to make web a first class citizen of a platform.

Thanks!