CROSSWALK PROJECT

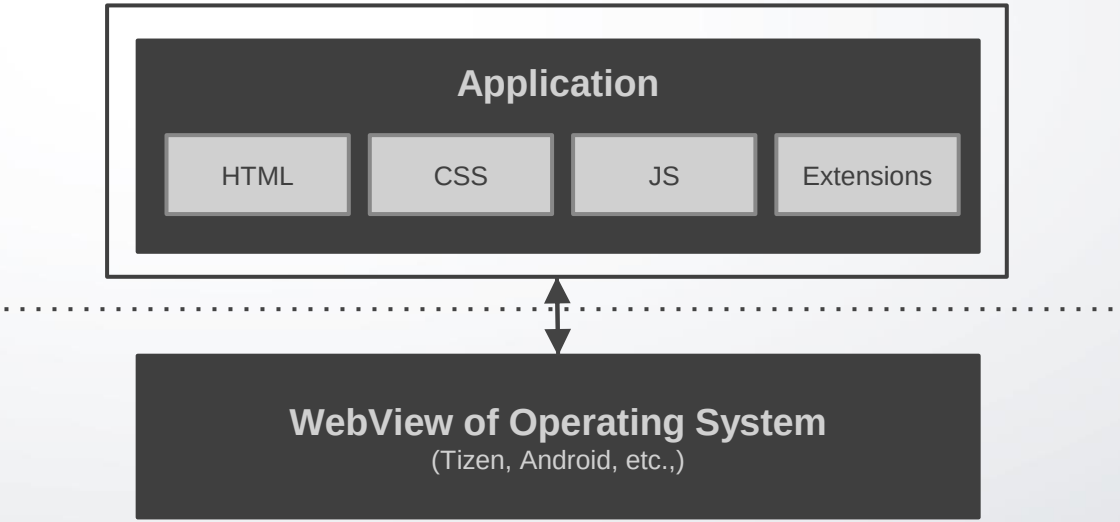
Crosswalk:

build world class hybrid mobile apps

Ningxin Hu

Intel

Today’s Hybrid Mobile Apps



State of Art

- HTML5 performance is not on par with native
- Limited set of API (especially system and hardware access) in comparison with native offerings. Furthermore, new APIs rolls out very slowly
- Deploying on Mobile OS is complicated
  - Multiple types of devices
  - Multiple versions of OS (sometimes very old)
- Current solutions:
  - Rely on the OS native WebView
  - Use PhoneGap/Cordova in combination with the above

-> Developer wants to target all the market, needs to stick with the common lowest feature set available

Crosswalk-Project

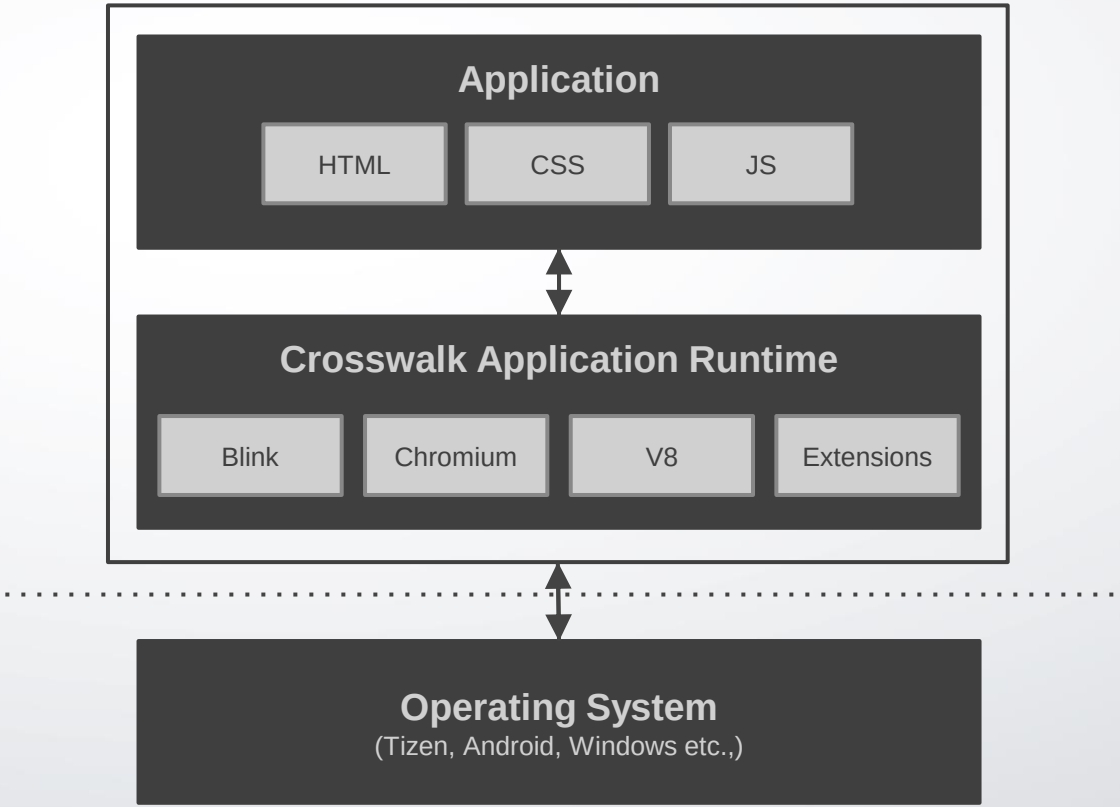
What is Crosswalk (1/2)

- Web-based runtime, which supports packaged apps
- Supported on Tizen and Android (powering 3k+ apps in Google Play).
- Experimentally supported on iOS and community supported on Linux and other major desktop platforms
- Apache Cordova compatible
- Landing zone for advanced Web APIs such as SIMD, WebCL, 3D camera etc.,
- Open source and BSD licensed -> FREE!!!!!

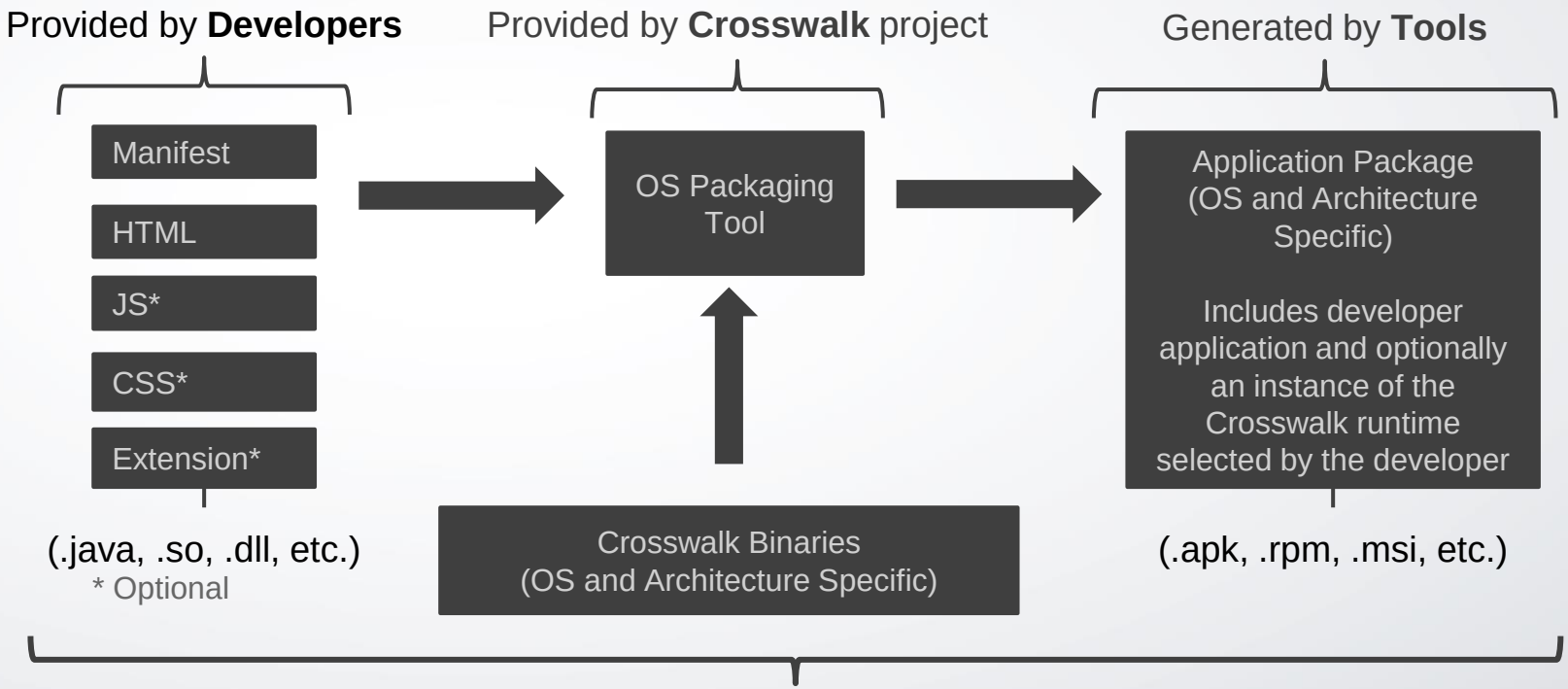
What is Crosswalk (2/2)

- Based on Chromium
  - Selected part of the project
  - On top of content/ layer
- Release often, every 6 weeks following new Chrome releases
  - ~1 week delay
- Optimized performance
  - Full screen web apps
- Mostly maintained by Intel, widely contributed by community

Crosswalk is a Runtime



Crosswalk Development Flow

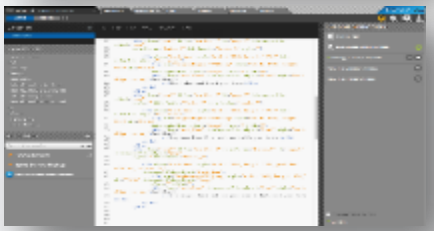


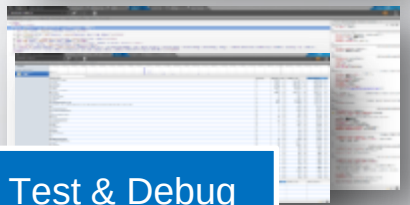
Available for use within these leading web app tools today

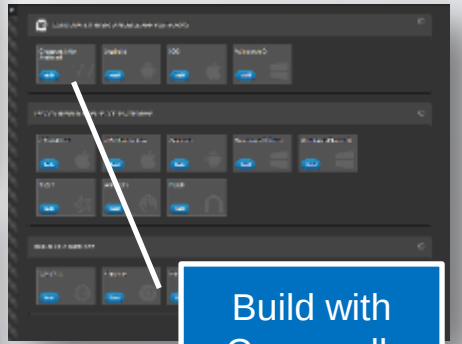
Intel® XDK · Chrome Apps for Mobile · Cordova · PhoneGap · Sencha · Famo.us · ionic · AppGyver

Crosswalk in Intel® XDK

Code → Test → Debug → Build







Build with Crosswalk

Test & Debug with Crosswalk

Crosswalk Features (1/3)

- WebAudio, WebGL, Canvas, Web Components, Service Worker, Responsive Design, Raw Sockets, Device Capabilities, WebRTC
- Support for Orientation Lock, Virtual Keyboard
- Remote debugging from your desktop
- You can trigger experimental Chrome features if you want to
- 3 channels : canary (daily build), beta and stable

Crosswalk Features (2/3)

- Cover hybrid use cases
  - mixed native components with a WebView in the middle
- Crosswalk has a embedding API for easy integration
- Integrate with the Intel XDK (IDE)
- Supports Cordova/PhoneGap
  - Access Cordova APIs
  - You can migrate your Cordova project to use Crosswalk and enjoy the benefits of a fresh new WebView
  - Cordova 4+ supports Crosswalk officially
- Support shared mode (no payload)

Crosswalk Features (3/3)

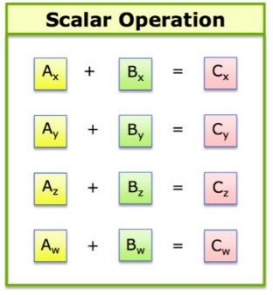
- SIMD.js : Vector parallel programming in Javascript (expose Intel SSE / ARM NEON)
- W3C Presentation API : applications can display content on multiple screens over wireless connection (expose wireless display)
- Extensions system : allows you to write native code and expose it in JavaScript
- W3C Manifest support for packaging your app
- Khronos WebCL : GPGPU programming on Web

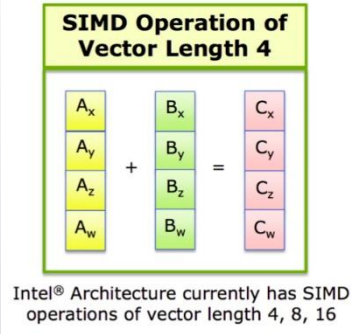
HTML5

HTML5 feature	Without the Crosswalk Project	With the Crosswalk Project
WebRTC	?	✔
WebGL	?	✔
Vibration API	?	✔
Presentation API	?	✔
`WebView` updates	?	✔

SIMD.js

- Perform operations on multiple data elements in parallel
  - SIMD = Single Instruction Multiple Data
  - Popular technique for accelerating computations in audio, graphics, physics simulations, etc.
  - In progress of being standardized for JavaScript with the target of ECMAScript 2016
- A known technique accessible to native application, and now brought to the web with cooperation from Intel, Google, Mozilla, Microsoft and ARM
- Available via Crosswalk Project, Chrome, Firefox and on Edge





WebCL

- Accelerate image/video processing and advanced physics for WebGL games
- WebCL enables web applications to harness GPU and multi-core CPU parallel processing, enabling significant acceleration for image and video processing and physics simulation and games
- WebCL 1.0 is a specification by the Khronos Group, defining a JavaScript API for the OpenCL standard for heterogeneous parallel computing

Crosswalk infrastructure (1/2)

- Follow the Chromium project practice
- Code hosted on GitHub and every patch is reviewed
- Bug tracking and requirements using Jira
- Project documentation hosted on Google Docs
- Every commit is tested in real time on all supported platforms
- Each channel is tested by QA (weekly) which reports about :
  - Performance
  - Feature testing
  - Regressions

Crosswalk infrastructure (2/2)

- We test over :
  - 8000+ Web APIs
  - 500 features
  - 60 use cases (e.g. packing an Hello World)
- On demand testing service
- We also have a custom Early Warning System on GitHub to test patches even before they are reviewed or committed
  - Also includes maintaining our forks of Chromium

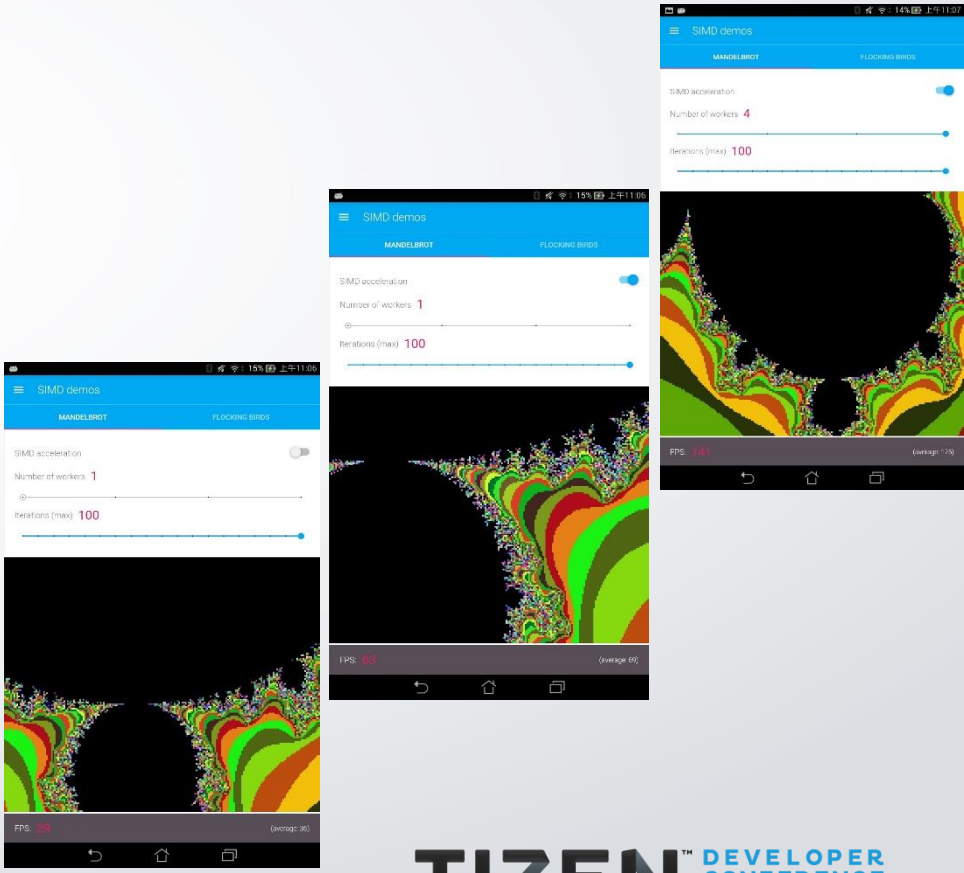
Summary

- Crosswalk enables stability, performance, and flexibility for developers
- Developers can deploy and control their applications
- You can finally use 2015 web platform technology cross mobile devices
- Crosswalk differentiates with advanced features

Crosswalk Differentiations

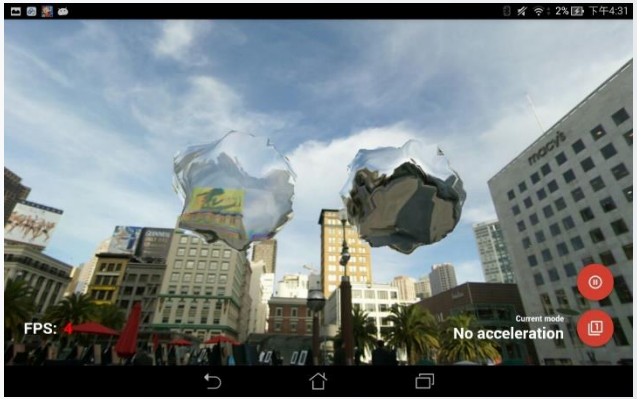
SIMD.js – Mandelbrot Acceleration

- Animating the Mandelbrot set is very computational intensive
- Web Workers can be used to calculate in parallel but there is some overhead in sending the data back
- SIMD can work on 4 data sets in parallel with no such overhead - and can be combined with Web Workers
- The demo is created using Polymer



WebCL – Deform

- A WebGL scene with two reflecting balls and a very computational intensive deform animation
- The Crosswalk Project enables WebCL on OpenCL enabled Android devices today
- WebCL enables the WebGL demo to harness GPU and multi-core CPU parallel processing, enabling significant acceleration for the deform transformation
- The demo is created using Polymer



Links

Web Site	www.crosswalk-project.org
Mailing list	crosswalk-dev@lists.crosswalk-project.org crosswalk-help@lists.crosswalk-project.org
IRC	#crosswalk @ freenode
WeChat	Crosswalk-project
Source Code	github.com/crosswalk-project
Download	https://crosswalk-project.org/documentation/downloads.html

Thanks!