GENIVI®

# Application Framework: Apertis Hands-on

2016-04-28

Robert Bosch Car Multimedia GmbH

Collabora Limited

# Apertis

APERTIS

# Apertis

- Free and open source, GNU/Linux-based platform for infotainment in automotive vehicles
- End-to-end implementation of an application centric solution for IVI
- Aiming at providing secure yet flexible environment that satisfy automotive OEM requirements and application developers' needs

# Apertis

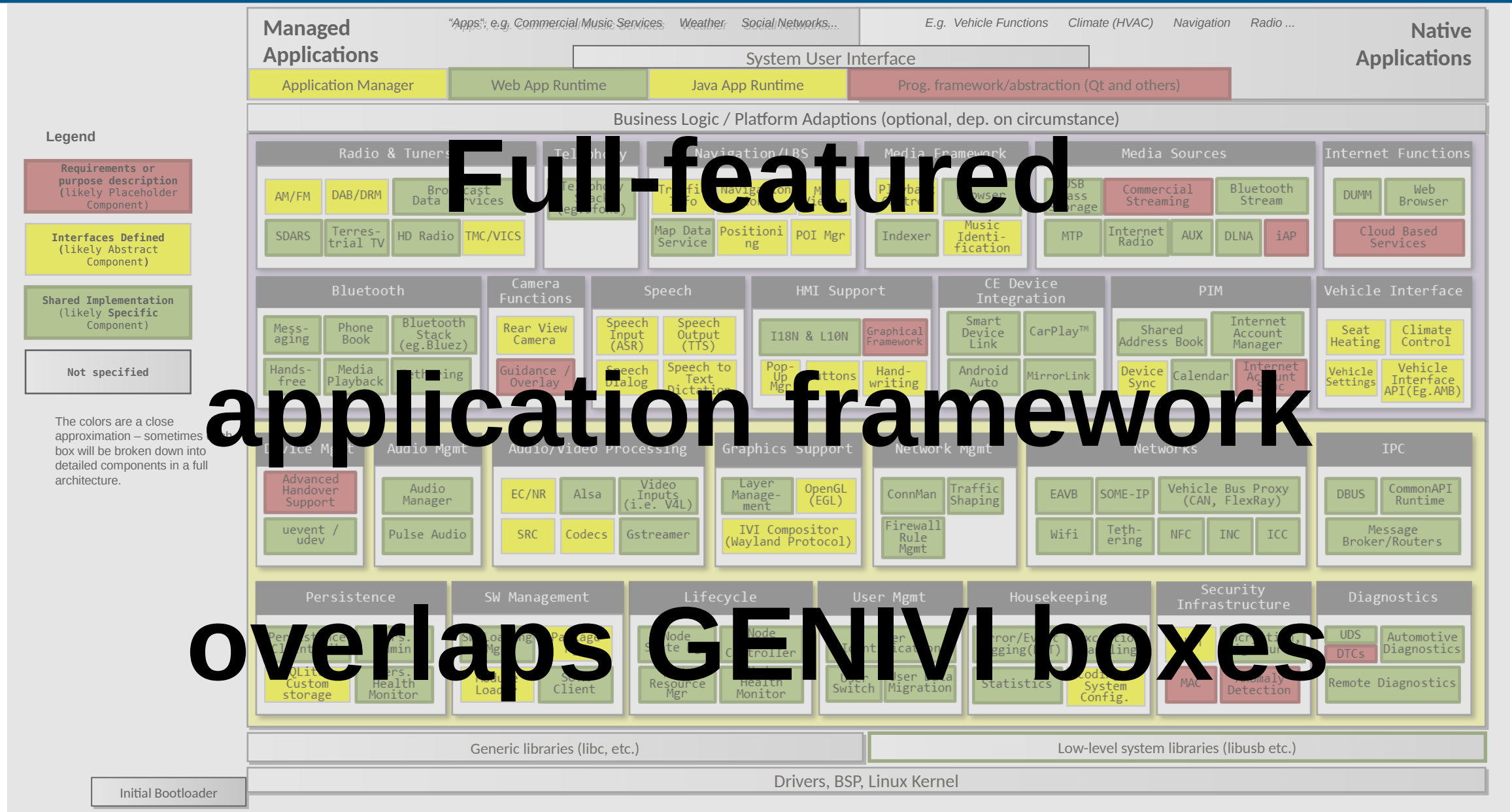
- GNU/Linux distribution derived from Debian/Ubuntu
- Product-specific images for ARM and Intel x86
- SDK virtual machine with full developer tools
- Developer portal and App Store for distribution

# Release early, Release often

While staying as stable as possible:

- Release every 3 months
- Rebase every 6 months

# Apertis




Full-featured application framework overlaps GENIVI boxes

# Applications

- Distinctions between applications
  - Built-in applications:
    - Core applications (required to run)
    - Pre-installed applications (selected by OEM, can not be uninstalled)
  - Applications coming from the application store
- Native and web applications
  - APIs currently primarily cater to proper C applications
- Services / Libraries
  - Provide functionalities to others

# Application framework for developers

- Leverage APIs from open source community
- Focuses on lowest common denominator for improving performance and memory consumption
- Roadmap items:
  - Adding JavaScript bindings for HTML5 applications is on the roadmap
  - Additional languages bindings (Python, C++, *etc.*)
  - Additional toolkits support

# Application development framework

**App Developers**

- Download SDK
- Install and Configure

**Install**

**Develop**

- Tutorials and API reference
- Develop app

- Simulator
- Development target
- GDB

**Test**

**Package**

- Package creation
- Publish on portal

SDK

**HTTPS**

Image & packages

community support

Publish

## Developer Portal

**Apertis Maintainers**

**APPSTORE**

**Deployment**

- Application security profile creation.
- Appstore versioning.
- App upload to store

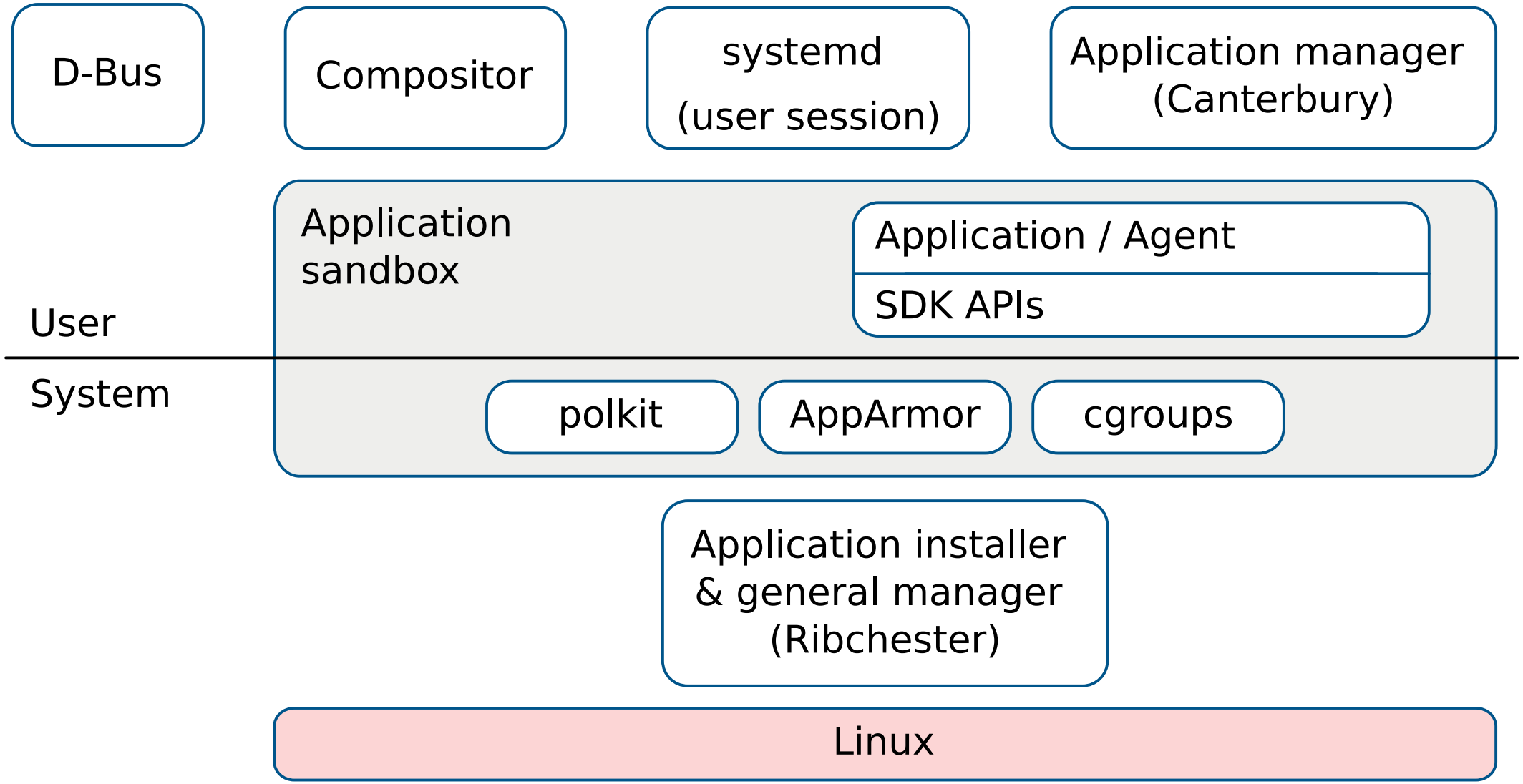
**Validation**

- App gesture verification
- App memory and CPU usage
- App's UI integrity
- User privacy verification

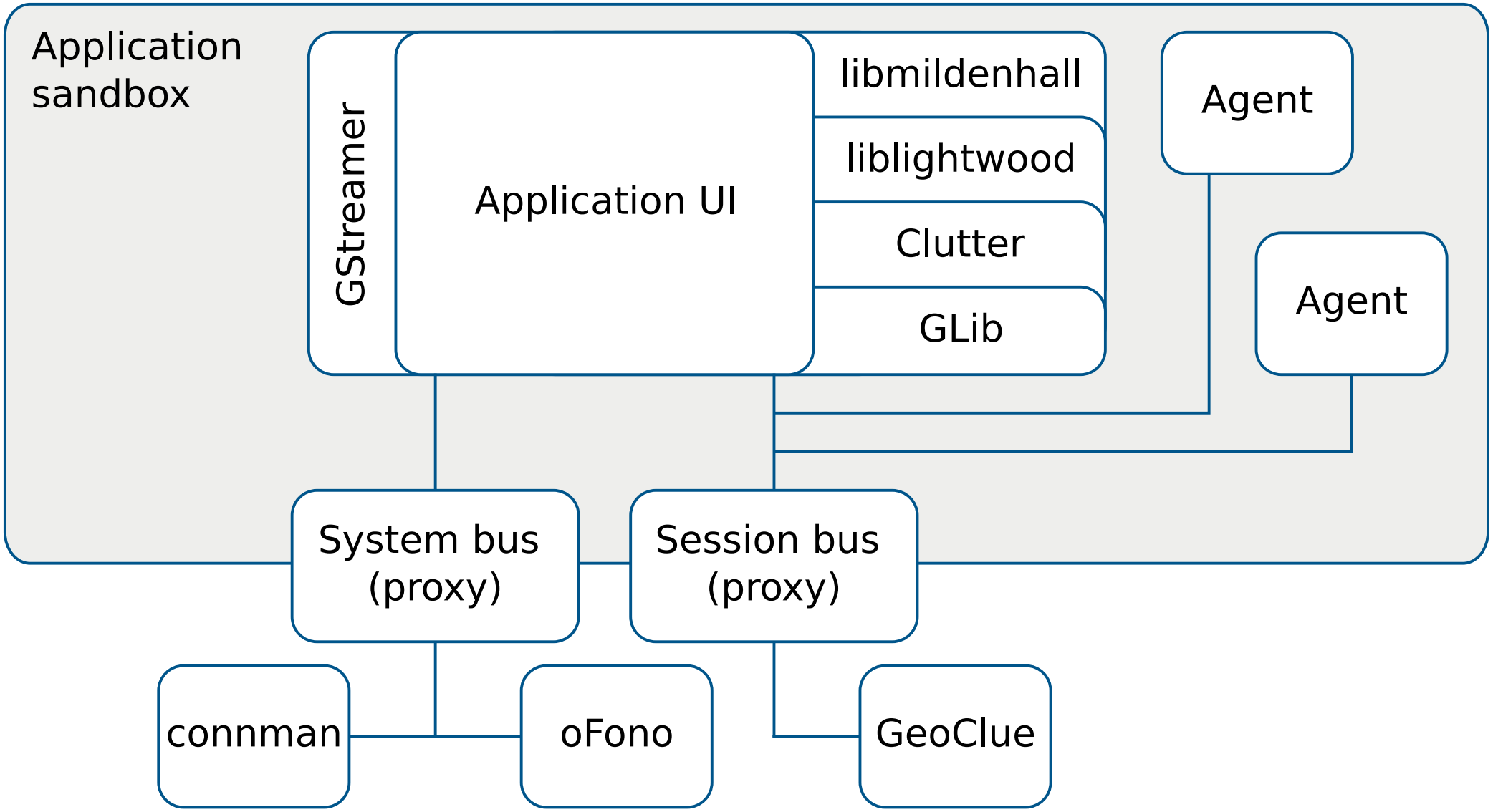
# Security

- **cgroups** provide resource control
- **polkit** provides policy decisions for services which talk to multiple applications
- **AppArmor** provides mandatory access control (MAC)

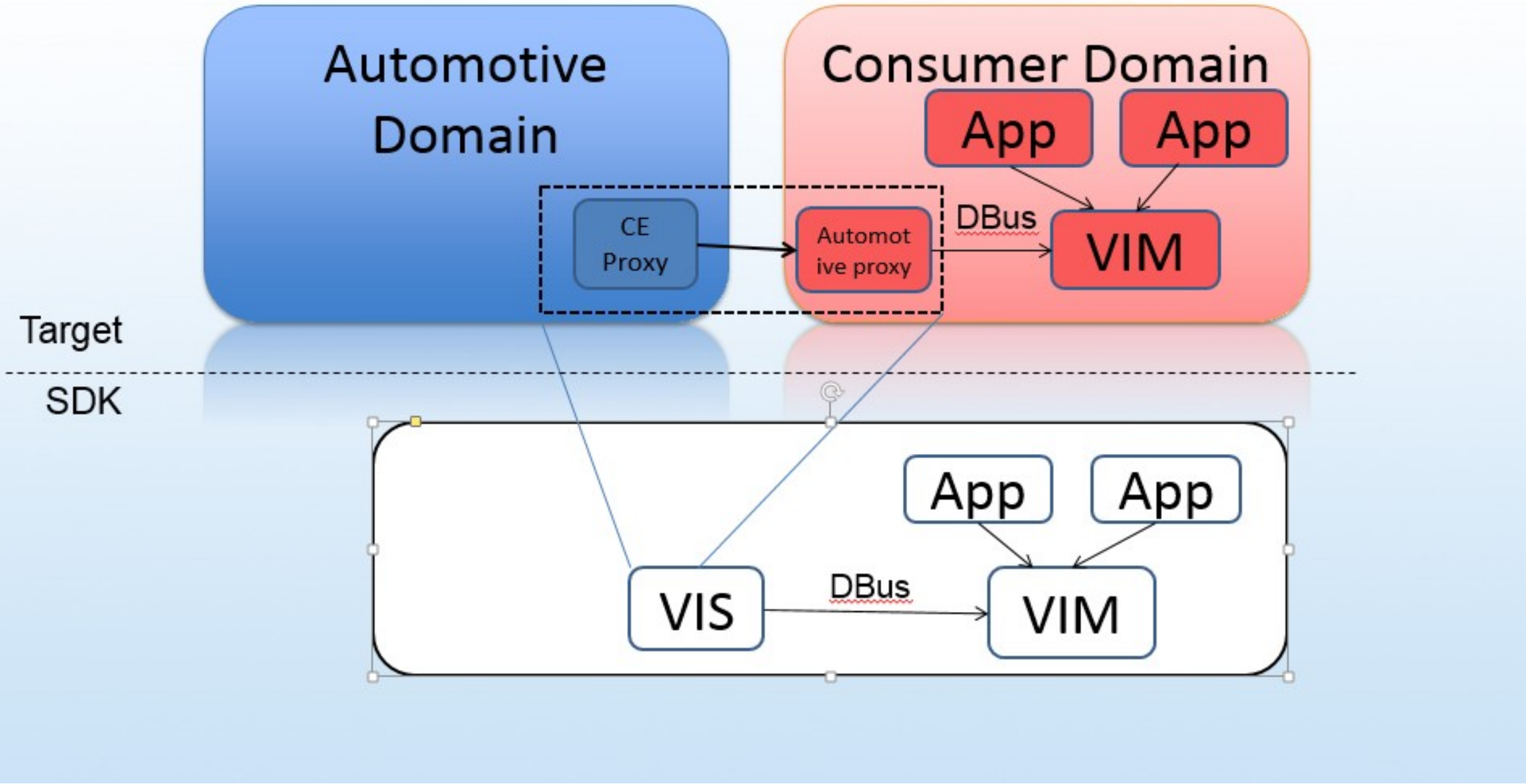
# Architecture



# Architecture



# Communication with the automotive domain



# SDK virtual machine

Your virtual machine image

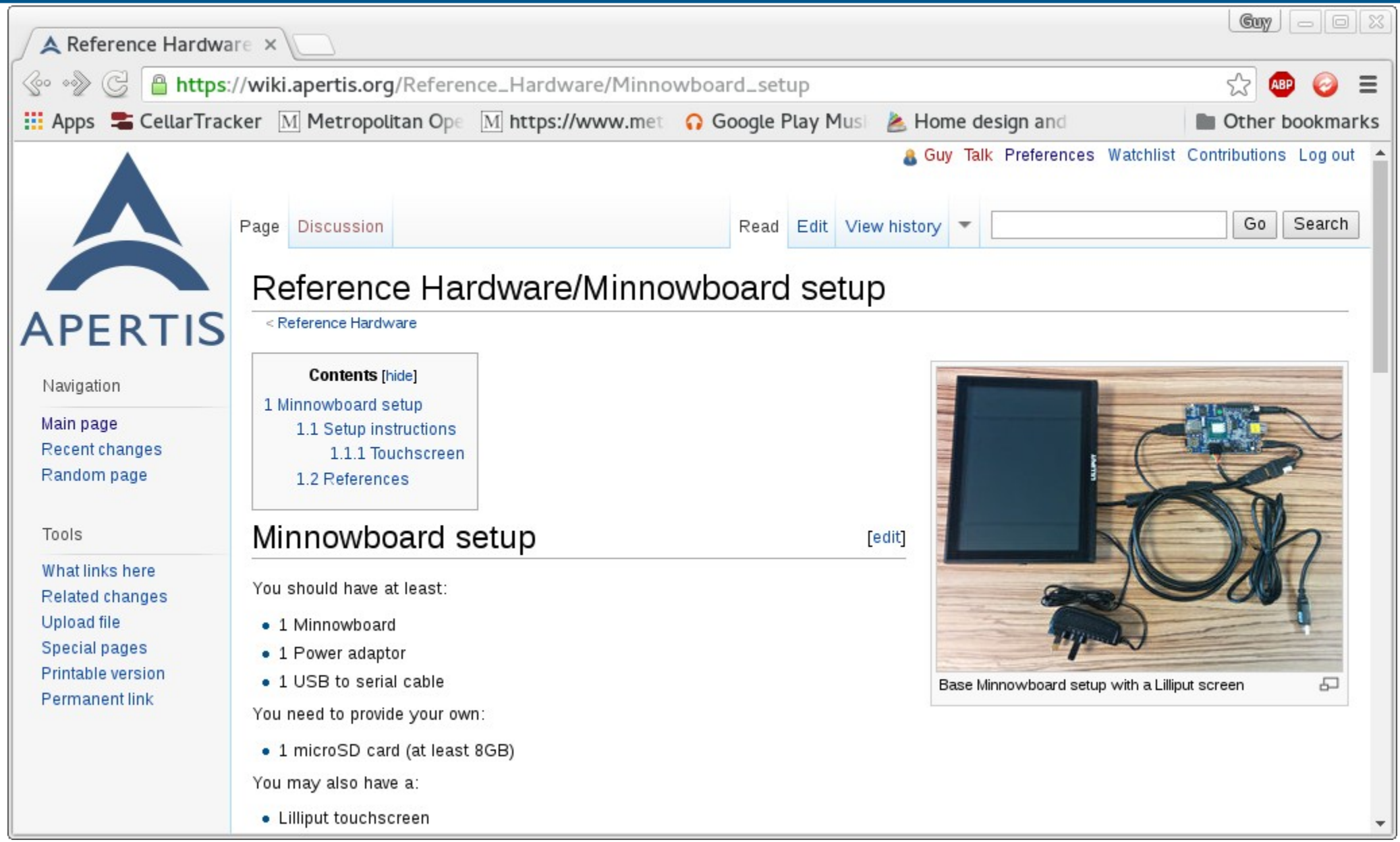
- Eclipse IDE with plugins
- Offline documentation
- Sample code
- Target simulator
- Packaging for App Store

# Application development support

## On-line resources

- Populated wiki
- API documentation
- Developer portal for app publication
- Developer forum, mailing lists, ...

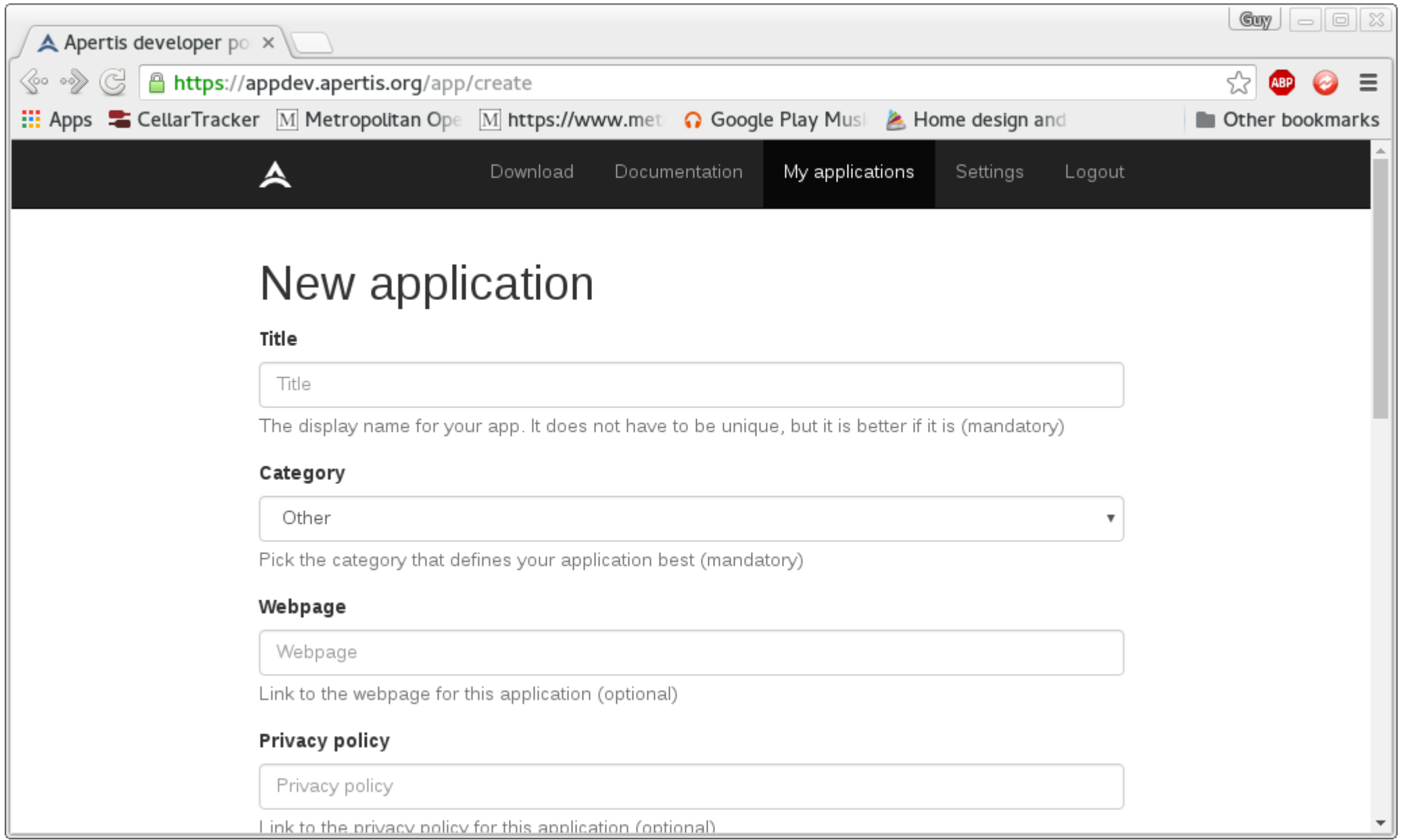
# Application development support



# Application development portal

- SDK download
- Latest documentation
- Upload applications for validation
- App revenue management
- Developer forum

# Application development portal



# Application validation

- License compliance
- Ethical and legal compliance
- Robustness checks
- UI/design compliance
- Version compatibility
- Security checks
- Backwards compatibility checks

# Updates to applications

- Updated versions can be uploaded through the development portal
- Updates through full validation as for first upload
- Deployed to App Store as soon as validation passes

# Apertis

APERTIS

# Hands-on session

1) Set up virtual machine in VirtualBox
2) Create your first Apertis application
3) Deploy application to simulator for testing
4) Deploy application to target for testing
5) Upload application to App Store

# Hands on demo

- Download SDK from appdev.apertis.org

https://Appdev.Apertis.org

# Apertis

APERTIS

# Thank you