Software Components for Secure Mobile Web Application Platforms

Patrik Persson & Björn Johansson

Ericsson Research
Mobile Platforms, Lund, Sweden

About us

- Ericsson Research in Lund, Sweden
  - Device security, application environments, ...
  - Working tightly with Ericsson Mobile Platforms (EMP)

- EMBRACE:
  Ericsson Mobile Browser Research
  And Cool Extensions
  - Prototype Widget-based terminal
  - EMP 3G platform, Linux kernel, Webkit, W3C-style Widgets

Motivation:
Separation of concerns

- Need separation of
  - Platform
  - Browser engine
  - Device APIs
  - Access control

- Need a vendor-neutral access control mechanism
  - Assuming policy defined by operator, manufacturer, community, or other

- Software component technology

ERICSSON

Why software components?

- Software component models
  - Separation of platform and application (separate address spaces possible)
  - Single entry point → centralized access control (method interception)
  - Can be fairly light-weight (e.g., COM/ECM)

- Interfaces described in IDL (interface description language)
  - Translated to glue code in JavaScript, C/C++, Java, ...
  - Language independence (with limitations)

COM/IDL translation example

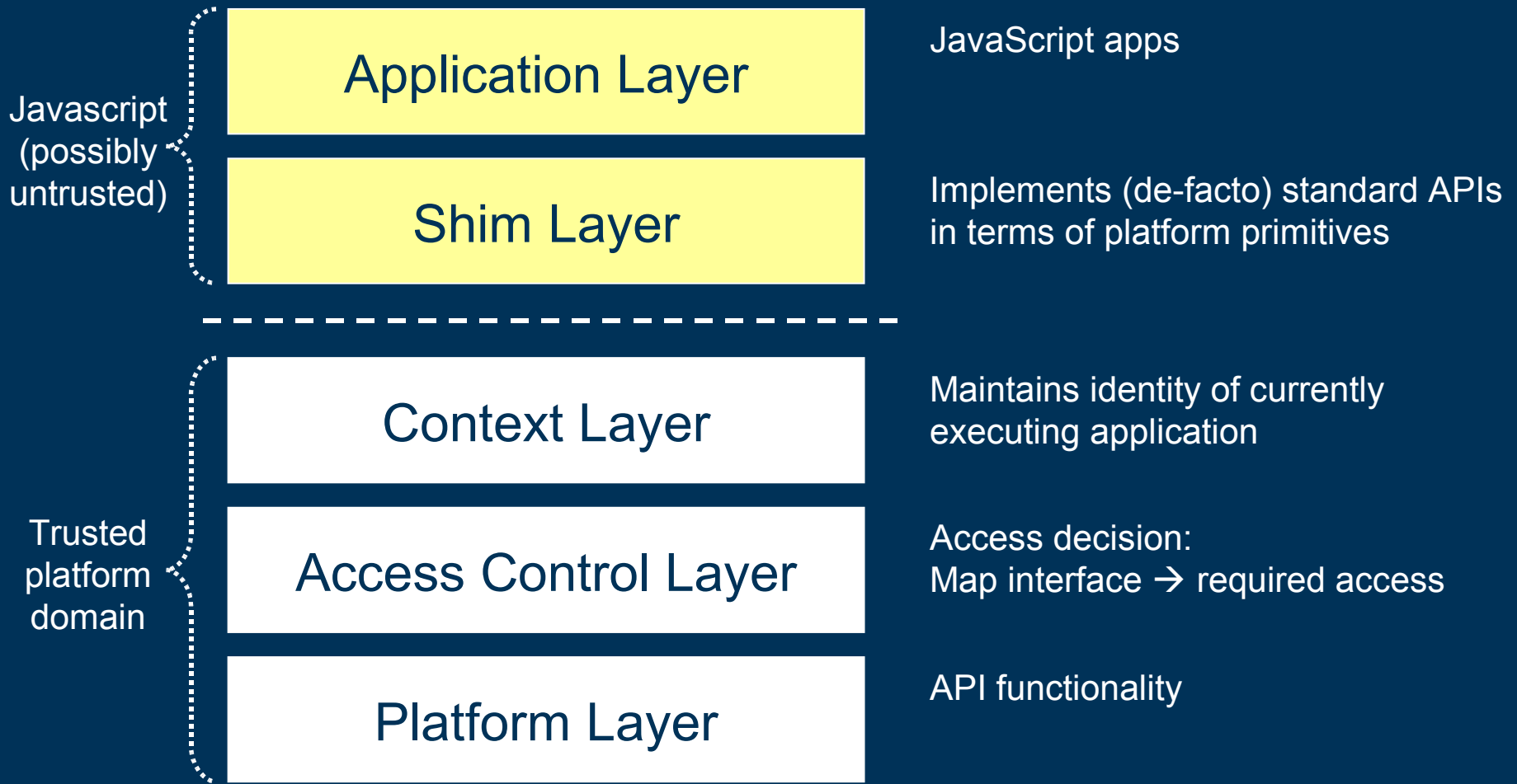**Interface Description Language (IDL)**

```
interface ICall {
  int  start(char * nbr);
  void stop(int session);
  void answer(int session);
  void reject(int session);
}
```

Generated proxy (JavaScript)

```
function ICall(...) {
  function start(nbr)      { ... };
  function stop(session)   { ... };
  function answer(session) { ... };
  function reject(session) { ... };
}
```

- Automatic IDL-to-JavaScript translator
  - COM interface instances → JavaScript proxy components
  - COM callback interfaces → JavaScript event handlers
- Some IDL limitations apply (e.g., regarding void*)

Basic architecture

Application Layer	JavaScript apps
Shim Layer	Implements (de-facto) standard APIs in terms of platform primitives

Javascript (possibly untrusted)

Context Layer	Maintains identity of currently executing application
Access Control Layer	Access decision: Map interface → required access
Platform Layer	API functionality

Trusted platform domain

Conclusions

- Advantages
    - Separation of concerns
    - Language independence
      (C/C++, Java, JavaScript, Python, Ruby, ...)
    - Single entry point → centralized access control
    - ECM (COM-like) proven in mobile devices

- Challenges
    - Maintaining run-time identity
    - Dynamically downloadable shim layers?
    - Performance & footprint
    - User experience

ERICSSON

TAKING YOU FORWARD