Dynamix: An Open Plug-and-Play Context Framework for Android

Darren Carlson and Andreas Schrader

Ambient Computing Group / Institute of Telematics

University of Lübeck, Germany

www.ambient.uni-luebeck.de

Motivation 1/2: The Explosive Rise of Mobile Computing

High Developer Incentives

✓ Familiar Languages & Tools — Mobile App Markets

Improved Device Capabilities

✓ Powerful Hardware — Inbuilt Comm/Sensors/Media

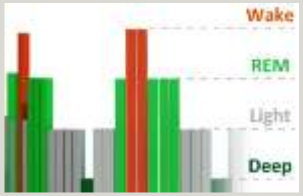
Basic Context Sensing

✓ APIs for accessing Location, Orientation, Sensor & User data, etc

Motivation 2/2: Context-awareness Challenges Remain

Advanced Context Sensing and Acting

Unproxied Sensor Networks



Biotelemetry Data

Indoor Positioning

User Activity

Social Proximity and Networks

Ad-hoc Interactions

External Sensors

Sensor Fusion

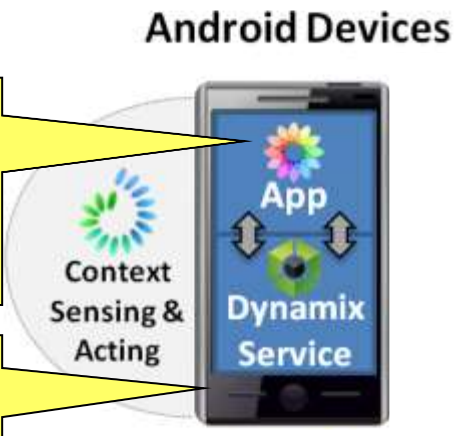
Others…

Wide-area Context Infrastructure

- Instrumentation scalability
- Multiple administrative domains (physical and virtual)
- Context sources/actuators not known at design time
- Lack of adaptive context middleware for mobile scenarios

Introducing Ambient Dynamix

Dynamix is a plug-and-play context framework that helps applications sense and adapt to the user's continuously evolving situation and requirements



Dynamix provides **simple means** for apps to request context support

Dynamix **adapts the user's device** to the environment **using plug-ins**

Includes a **scalable infrastructure** for sharing plug-ins

Plug-ins are provisioned to the device **at runtime** (network or file system)

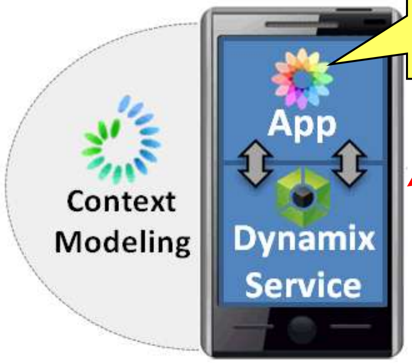
Domain experts create **context plug-ins**

Community-based approach with 3rd party API support

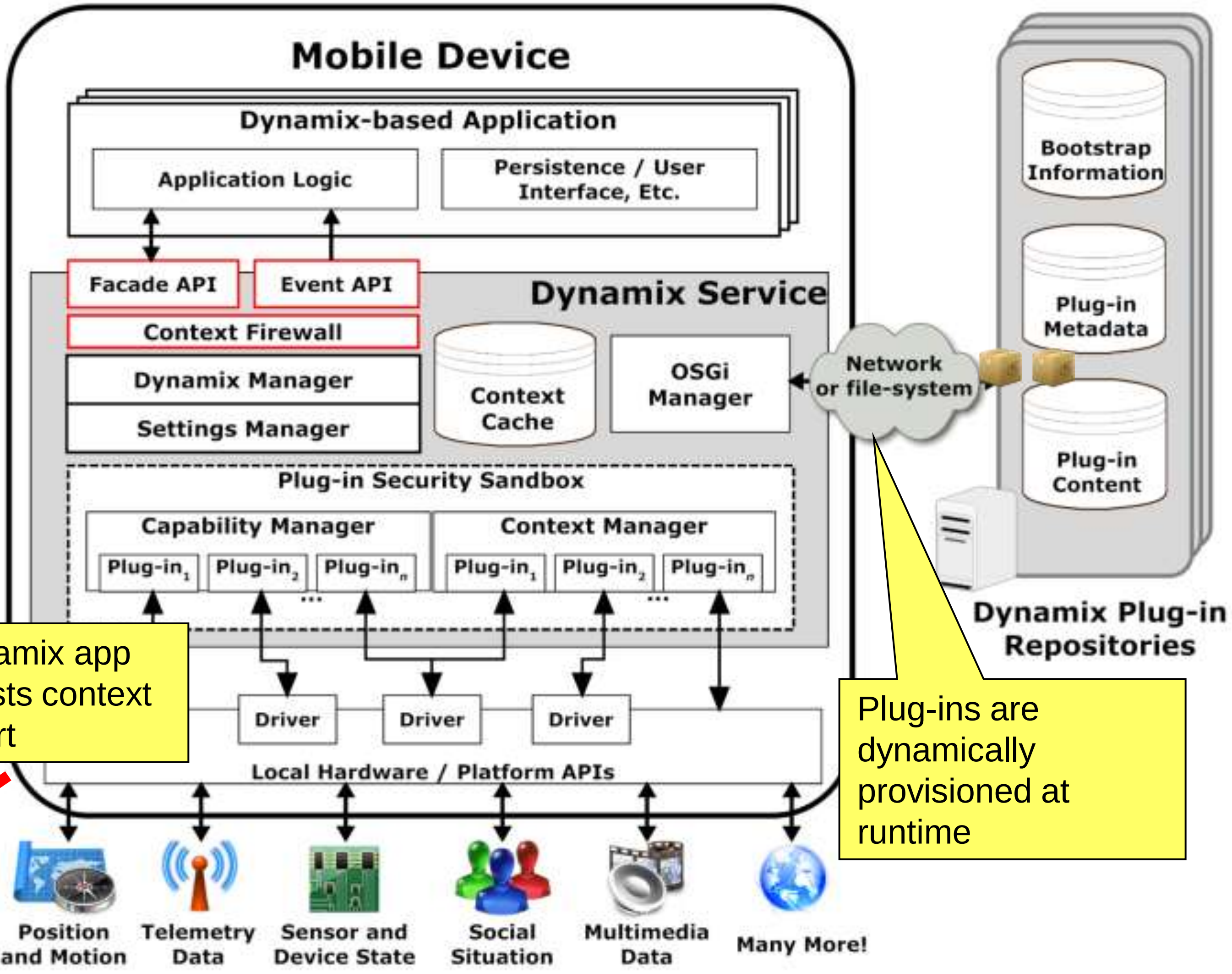
Ambient Dynamix
ambientdynamix.org

Overview of the Dynamix Framework

Context domain expertise is packaged into sharable plug-ins

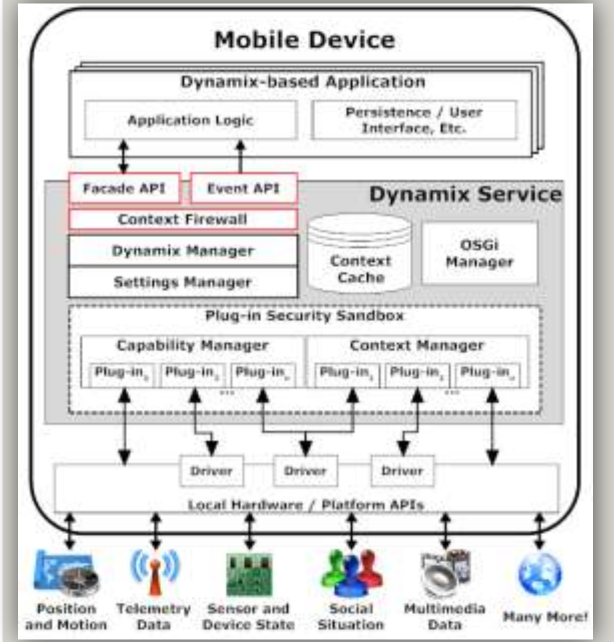


A Dynamix app requests context support



Plug-ins are dynamically provisioned at runtime

Dynamix Framework Features



- Runs as a lightweight **background service** on a user's unmodified Android-based device.
- Apps use simple **Facade and Event APIs** to request context support and receive context events.
- Performs context interactions using a tailored set of plug-ins, which are **dynamically provisioned** to the device during runtime *(from the network or local file-system)*.
- Supports **ad-hoc interactions** with discovered resources.
- Sends context information to apps using plain old Java objects (**POJOs**) or string-encoded formats.
- Supports parallel plug-in installations, automatic updating, event caching, and power management.
- Utilizes an **embedded OSGi Framework** to manage Dynamix plug-ins internally.
- Features a **Plug-in Security Sandbox,** which provides managed access to sensitive services and

Open Community Collaboration

1

Open Plug-in SDK

Context-domain experts use the Open Plug-in SDK to create Dynamix plug-ins

Plug-ins can be published using public or private repositories

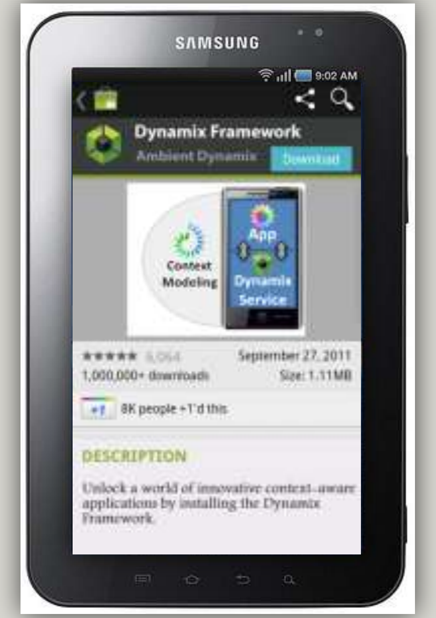
2

Open App SDK

App developers use the Open App SDK to create Dynamix apps

Apps can be deployed from any Android market or elsewhere

3



Dynamix Framework
Ambient Dynamix

September 27, 2011
1,000,000+ downloads
Size: 1.11MB
8K people +1'd this

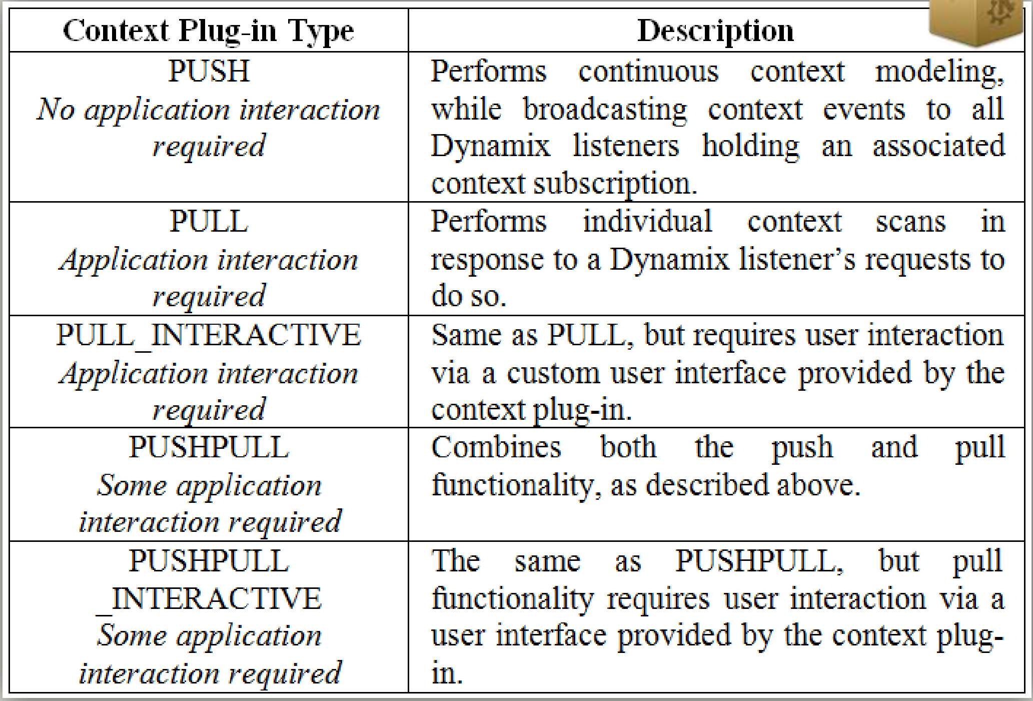
DESCRIPTION
Unlock a world of innovative context-aware applications by installing the Dynamix Framework.

The end-user installs the Dynamix Framework once

Users can then download and run Dynamix apps

Context Plug-in Development Overview

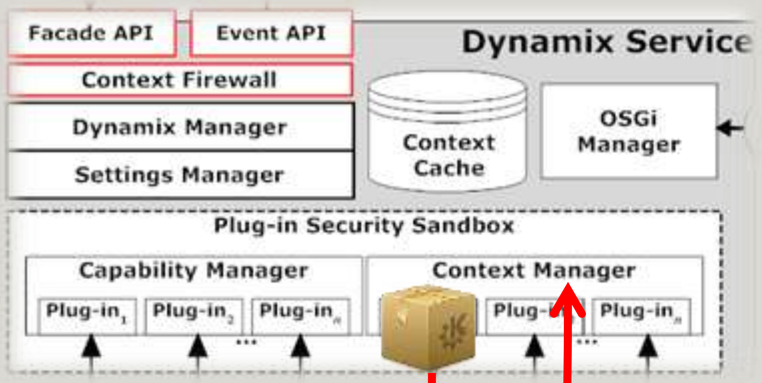
Open Plug-in SDK

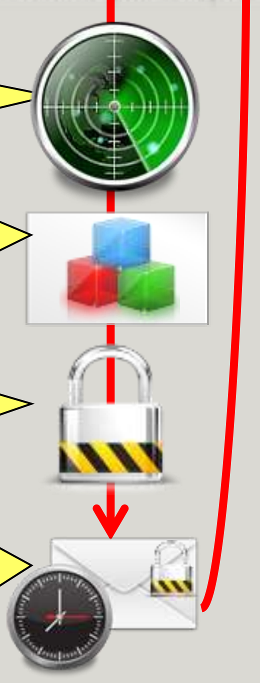


Context Plug-in Type	Description
PUSH *No application interaction required*	Performs continuous context modeling, while broadcasting context events to all Dynamix listeners holding an associated context subscription.
PULL *Application interaction required*	Performs individual context scans in response to a Dynamix listener's requests to do so.
PULL_INTERACTIVE *Application interaction required*	Same as PULL, but requires user interaction via a custom user interface provided by the context plug-in.
PUSHPULL *Some application interaction required*	Combines both the push and pull functionality, as described above.
PUSHPULL_INTERACTIVE *Some application interaction required*	The same as PUSHPULL, but pull functionality requires user interaction via a user interface provided by the context plug-in.

Includes base classes for a variety of plug-in types

Developers can release custom data-types as standard JAR file, which are used by app developers.





Context Sensing or Acting

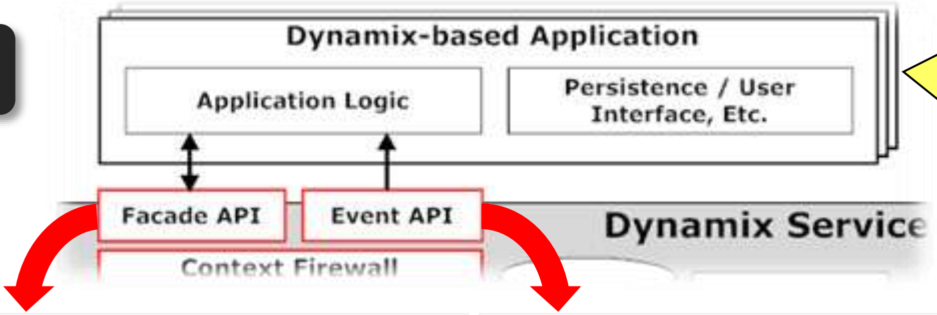
Context Representation (POJO or String)

Privacy Risk Tagging

Context Event Provisioning (includes time-stamp and validity duration metadata)

Dynamix App Development Overview

Open App SDK



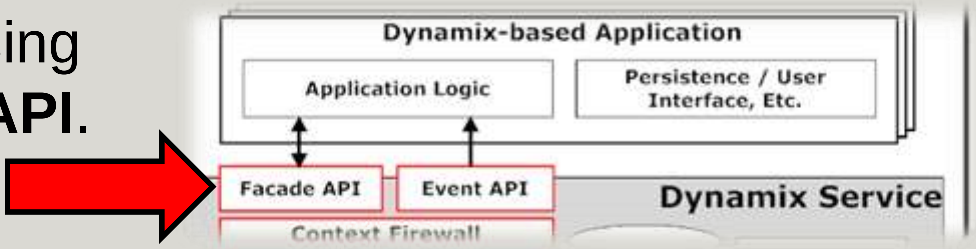
Apps include a single Dynamix JAR on their build path

Facade API Method Summary	
void	addDynamixListener(IDynamixListener listener) Registers the listener to receive Dynamix events.
void	openSession() Indicates that the calling application wishes to open a session with the Dynamix Service.
void	addContextSubscription(IDynamixListener listener, String contextType) Adds a context subscription for the specified listener and context type.
String	requestContextScan(IDynamixListener listener, String pluginId, String contextType) Requests a dedicated context scan using the specified plug-in and context type.
void	resendCachedContextEvents(IDynamixListener listener, int previousMills) Resends the context events that have been cached for the listener.
Other methods have been omitted for brevity…	

Event API Method Summary	
void	onDynamixListenerAdded(String listenerId) Indicates that the Dynamix listener has been added.
void	onSecurityAuthorizationGranted() Notification that the application has been granted security authorization by the Dynamix Service.
void	onSessionOpened(String sessionId) Notification that a Dynamix session has been opened.
void	onContextSubscriptionAdded(ContextPluginInformation plugin, String contextType) Notifies the listener that a context subscription for the given context type has been added.
void	onContextEvent(ContextEvent event) Notification of an incoming context event.
Other events have been omitted for brevity…	

Setting Up Context Support

Apps request context support using the Dynamix Service's **Façade API**.

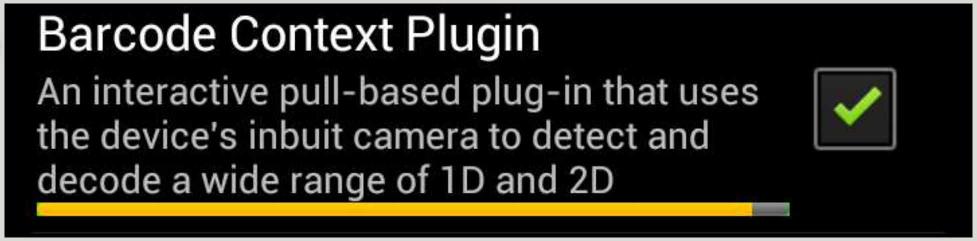


1

```
dynamix.addContextSubscription(dynamixCallback,
          "org.ambientdynamix.contextplugins.barcode");
```

Apps add context subscriptions for required context types

2



Barcode Context Plugin

An interactive pull-based plug-in that uses the device's inbuit camera to detect and decode a wide range of 1D and 2D

Dynamix downloads and installs associated plug-ins in the background

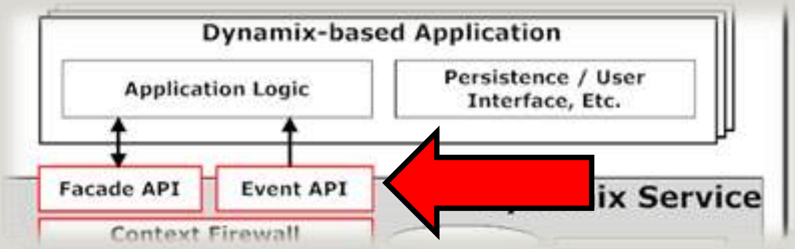
3

```
dynamix.contextRequest(dynamixCallback,
          "org.ambientdynamix.contextplugins.barcode",
          "org.ambientdynamix.contextplugins.barcode");
```

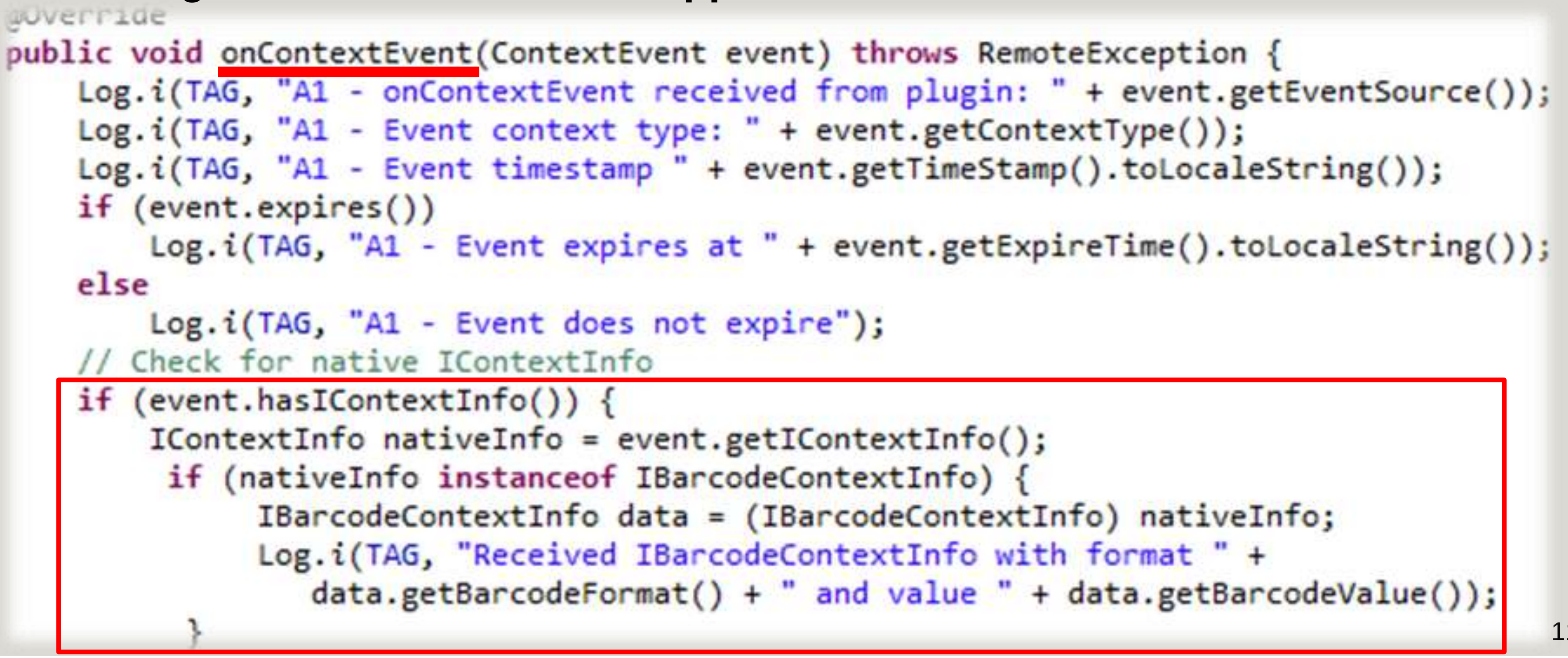
If necessary, apps trigger context requests (scans or interactions)

Handling Context Events

Context events are sent to apps using the Dynamix Service's **Event API.**

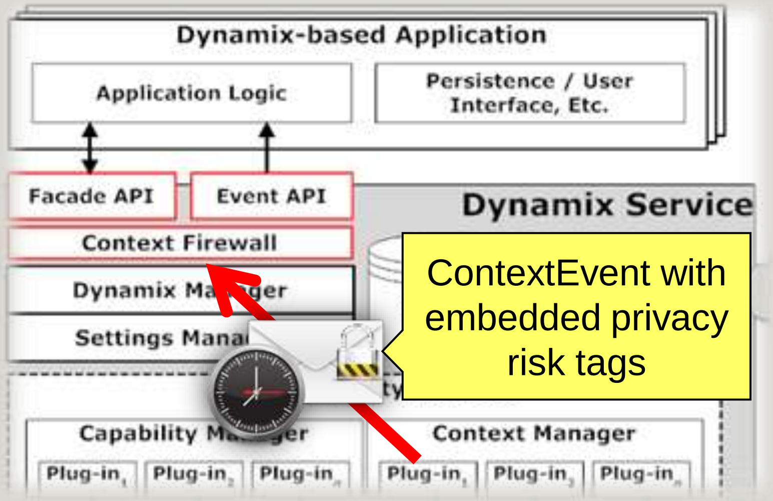


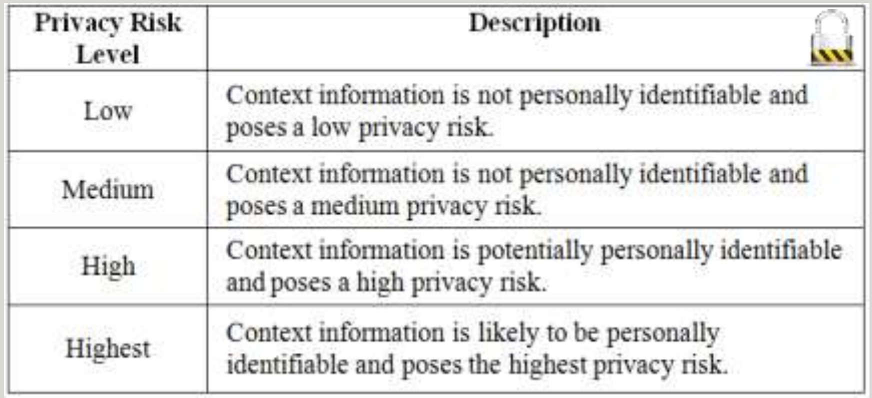
Receiving Context Events In the App:



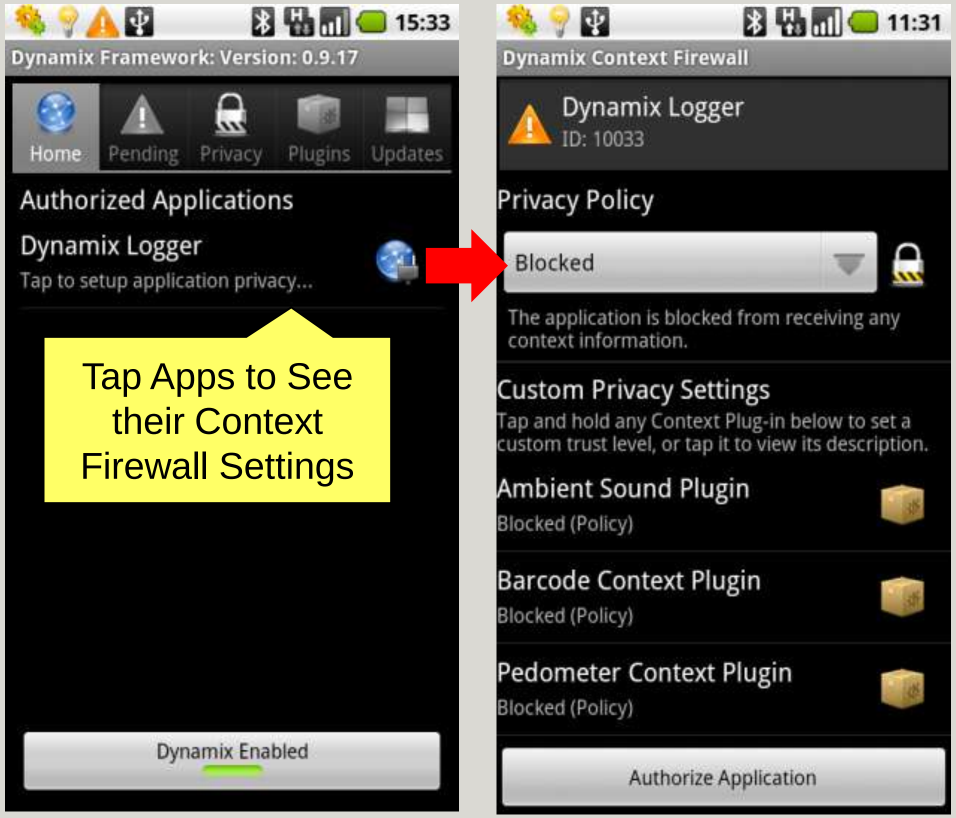
```
@Override
public void onContextEvent(ContextEvent event) throws RemoteException {
    Log.i(TAG, "A1 - onContextEvent received from plugin: " + event.getEventSource());
    Log.i(TAG, "A1 - Event context type: " + event.getContextType());
    Log.i(TAG, "A1 - Event timestamp " + event.getTimeStamp().toLocaleString());
    if (event.expires())
        Log.i(TAG, "A1 - Event expires at " + event.getExpireTime().toLocaleString());
    else
        Log.i(TAG, "A1 - Event does not expire");
    // Check for native IContextInfo
    if (event.hasIContextInfo()) {
        IContextInfo nativeInfo = event.getIContextInfo();
         if (nativeInfo instanceof IBarcodeContextInfo) {
            IBarcodeContextInfo data = (IBarcodeContextInfo) nativeInfo;
            Log.i(TAG, "Received IBarcodeContextInfo with format " +
               data.getBarcodeFormat() + " and value " + data.getBarcodeValue());
          }
```

Protecting User Privacy with the Context Firewall





Privacy Risk Level	Description
Low	Context information is not personally identifiable and poses a low privacy risk.
Medium	Context information is not personally identifiable and poses a medium privacy risk.
High	Context information is potentially personally identifiable and poses a high privacy risk.
Highest	Context information is likely to be personally identifiable and poses the highest privacy risk.



Dynamix Framework: Version: 0.9.17

Authorized Applications
Dynamix Logger
Tap to setup application privacy...

Dynamix Context Firewall
Dynamix Logger
ID: 10033

Privacy Policy
Blocked
The application is blocked from receiving any context information.

Custom Privacy Settings
Tap and hold any Context Plug-in below to set a custom trust level, or tap it to view its description.

Ambient Sound Plugin
Blocked (Policy)

Barcode Context Plugin
Blocked (Policy)

Pedometer Context Plugin
Blocked (Policy)

Dynamix UI: Context Firewall Management

Apps Can Leverage a Broad Range of Dynamix Plug-ins

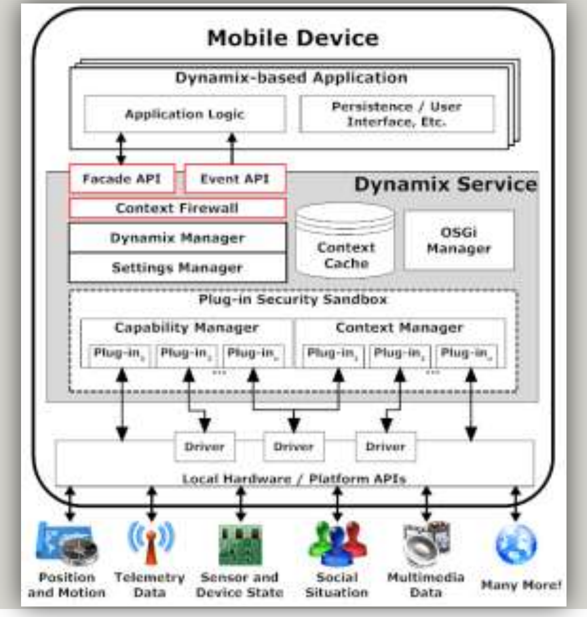
Context Plug-in	
Power-aware Location, NFC, Beacon, Orientation, and Photodetector Plug-ins	✓
ArtNet Plug-in (ad-hoc discovery/control of DMX automation equipment)	✓
Acoustic Fingerprint Plug-in (Native Code Integration)	
Speech Recognition Plug-in	
Sound Pressure Level Plug-in (Ambient Sound Detector)	
Sleep State Plug-in (Zeo Mobile)	
Heart-rate Biotelemetry Plug-in (Zephyr Sensors)	
OpenSocial Profile Data with Sensor-network Monitoring Plug-in (SmartAssist)	✓
Barcode Scanner Plug-in (ZXing port)	
Air Quality Monitor Plug-in (Ozone Levels and Pollen Count)	
Weight and BMI Measurement Plug-)	

Many more plug-ins in development!

Implementation and Evaluation 1/2

Dynamix Framework

- Comprehensive OSGI-based Android prototype
- Plug-in and App SDKs
- Repository architecture
- Website and documentation
- Tested on many popular Android device types



Plug-in Development

- 15 initial plug-ins (more soon)
- Range of context domains and semantics (push vs. pull)
- Each verified as deployable over-the-air at runtime

App Development

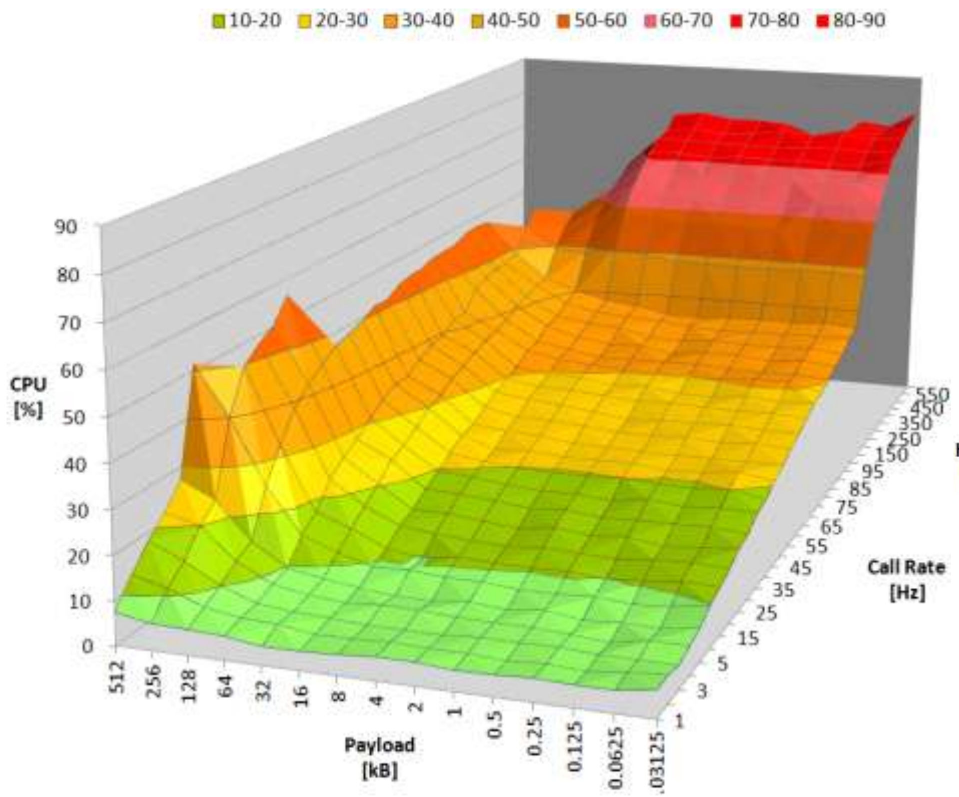
- Six initial prototype apps
- Additional apps are being developed

Dynamix Apps
Dynamix Logger
Medication reminder system
Product information and reviews
Bike Wars! Social exercise app
Heart rate visualization *(Processing graphics engine)*
Ambient campus information
Sound of the City *(in development)*
More soon!

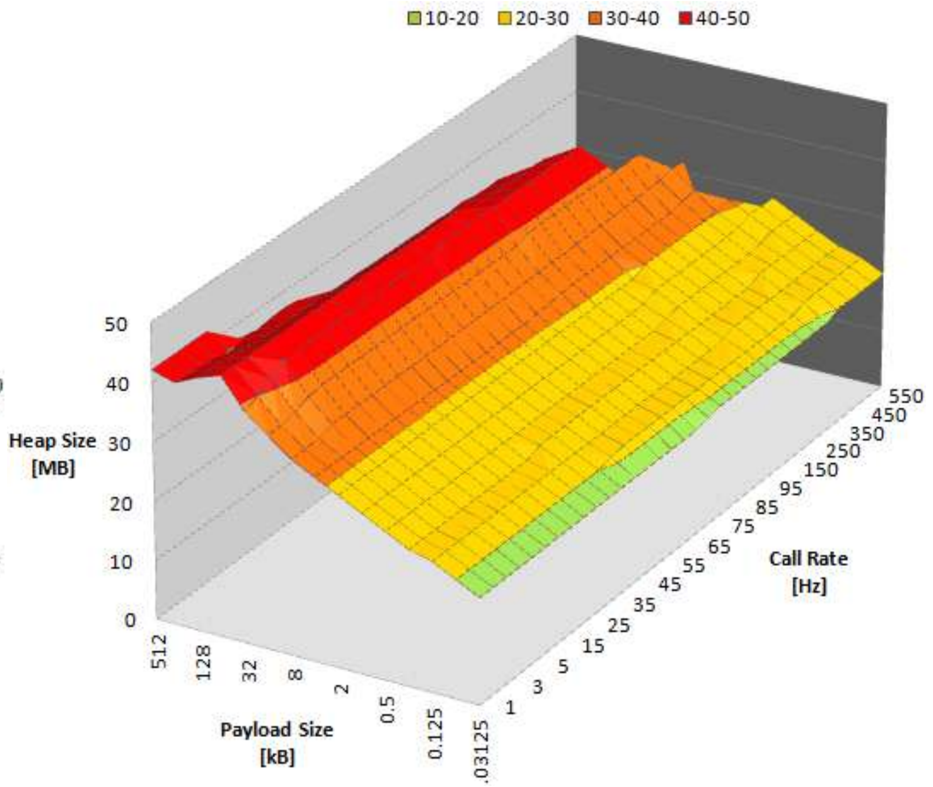
AmbientWeb Extension

- Exposes full Dynamix functionality to *browser-based* Web clients
- Drop-in JavaScript libraries
- Includes Wellness App demo, created for the IoT 2012 Challenge

Implementation and Evaluation 2/2



Total CPU% for Various Payloads and Call Rates



Heap Size for Various Payload Sizes and Call Rates

Dynamix exhibits **linear performance characteristics (CPU and Heap)** for typical context scanning rates and event payload sizes

Help Us Make Dynamix Even Better!

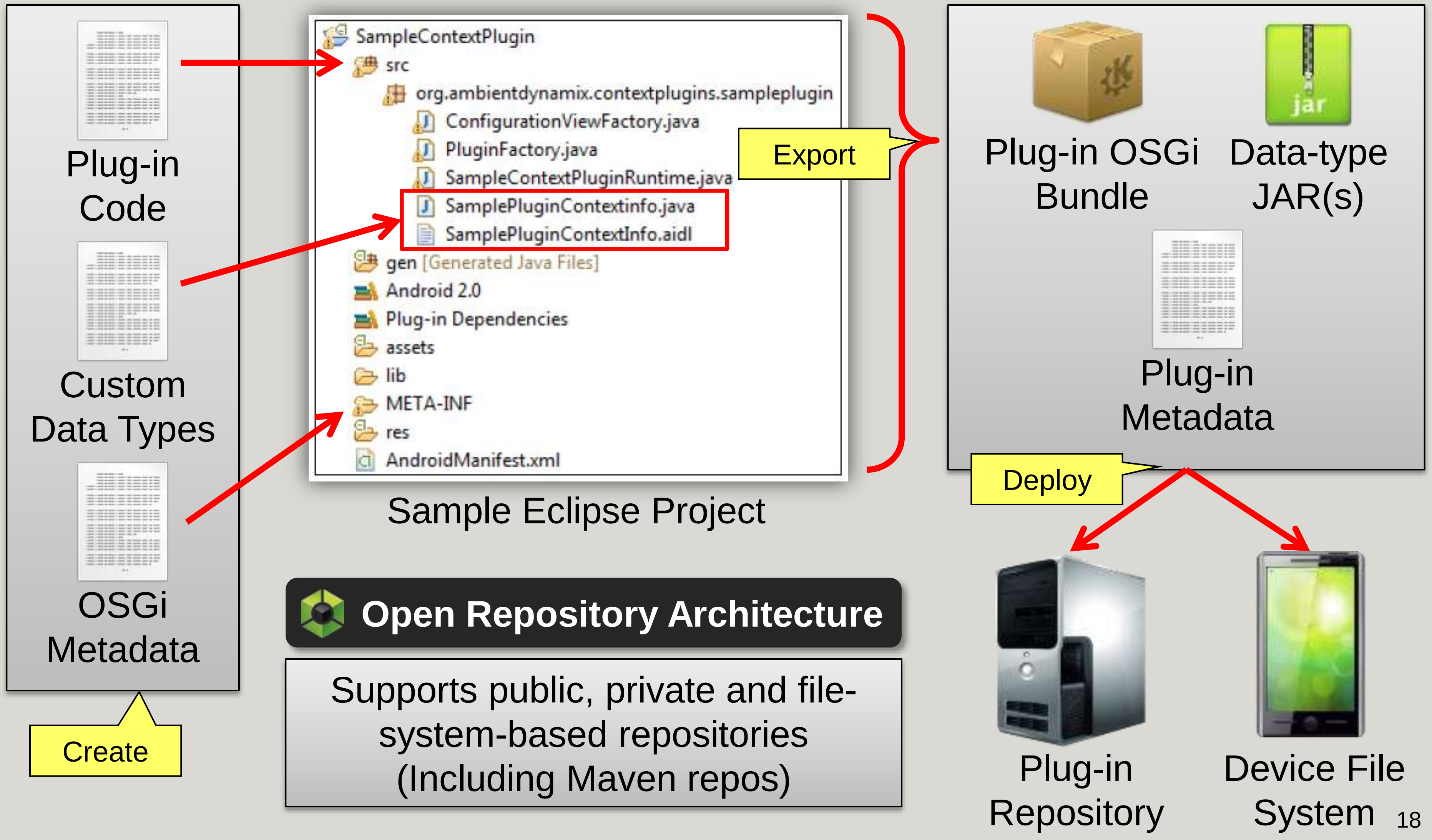
Uncover a world of context information...
with only a few lines of code

- Dynamix is free, open-source, and looking for contributors!
- Visit **ambientdynamix.org** for details and documentation.
- To access the developer kits, join the public beta!
  Email: carlson@itm.uni-luebeck.de for access.

Thank You!

Q&A

Context Plug-in Packaging and Deployment



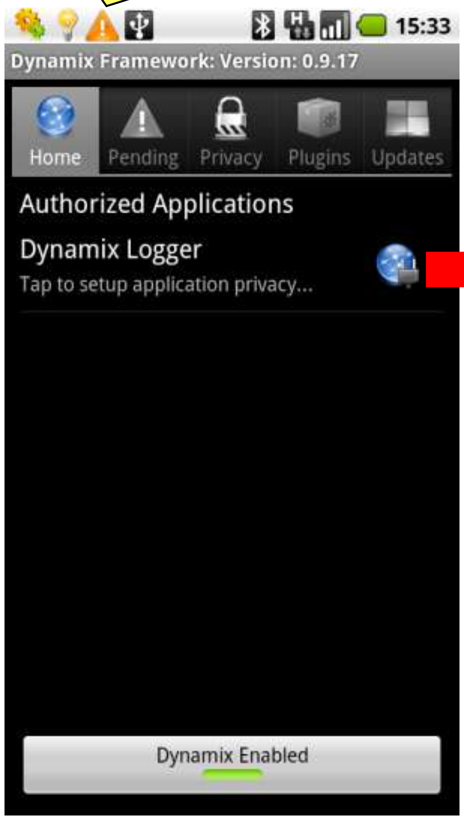
```
SampleContextPlugin
  src
    org.ambientdynamix.contextplugins.sampleplugin
      ConfigurationViewFactory.java
      PluginFactory.java
      SampleContextPluginRuntime.java
      SamplePluginContextinfo.java
      SamplePluginContextInfo.aidl
  gen [Generated Java Files]
  Android 2.0
  Plug-in Dependencies
  assets
  lib
  META-INF
  res
  AndroidManifest.xml
```

Open Repository Architecture

Supports public, private and file-system-based repositories (Including Maven repos)

Dynamix from the End-user’s Perspective

Android Notifications



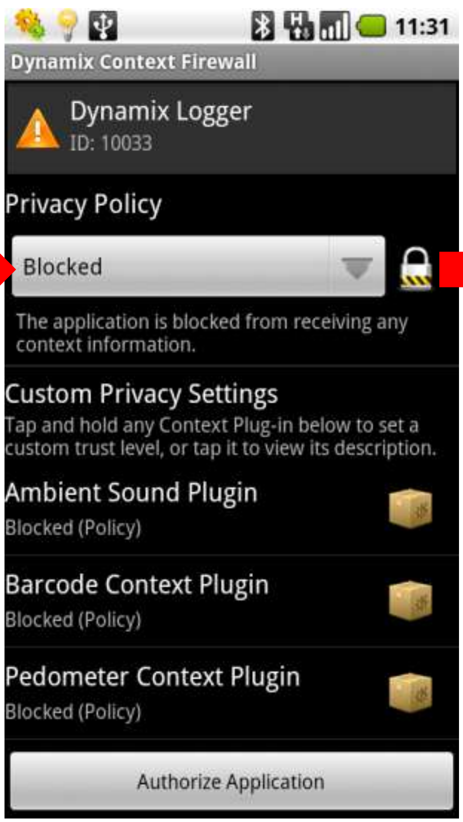
Dynamix Framework: Version: 0.9.17

Authorized Applications

Dynamix Logger
Tap to setup application privacy...

Dynamix Enabled

The Home Tab



Dynamix Context Firewall

Dynamix Logger
ID: 10033

Privacy Policy

Blocked

The application is blocked from receiving any context information.

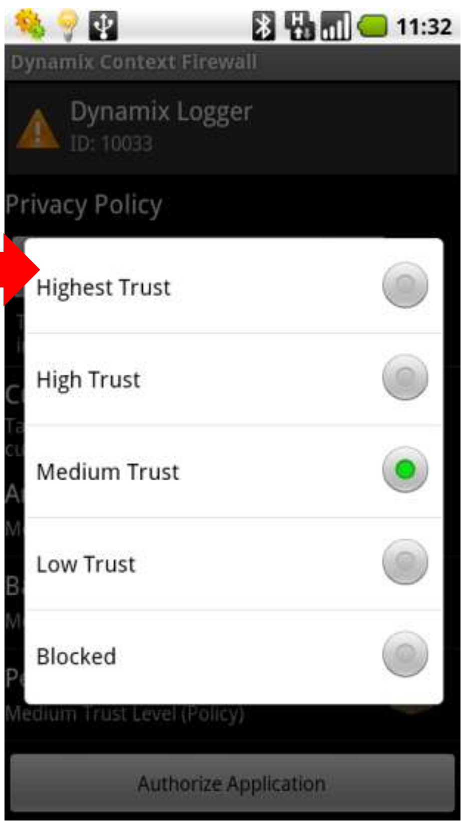
Custom Privacy Settings
Tap and hold any Context Plug-in below to set a custom trust level, or tap it to view its description.

Ambient Sound Plugin
Blocked (Policy)

Barcode Context Plugin
Blocked (Policy)

Pedometer Context Plugin
Blocked (Policy)

Authorize Application



Dynamix Context Firewall

Dynamix Logger
ID: 10033

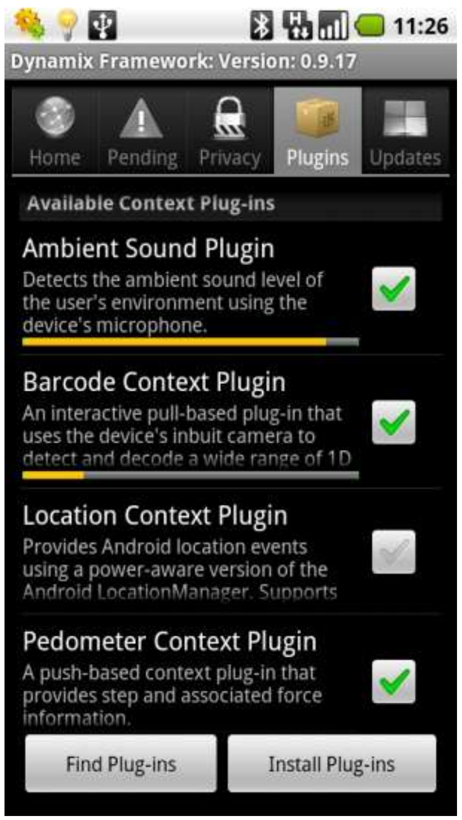
Privacy Policy

- Highest Trust
- High Trust
- [✓ Medium Trust]
- Low Trust
- Blocked

Medium Trust Level (Policy)

Authorize Application

The Context Firewall



Dynamix Framework: Version: 0.9.17

Available Context Plug-ins

Ambient Sound Plugin ✓
Detects the ambient sound level of the user's environment using the device's microphone.

Barcode Context Plugin ✓
An interactive pull-based plug-in that uses the device's inbuilt camera to detect and decode a wide range of 1D

Location Context Plugin
Provides Android location events using a power-aware version of the Android LocationManager. Supports

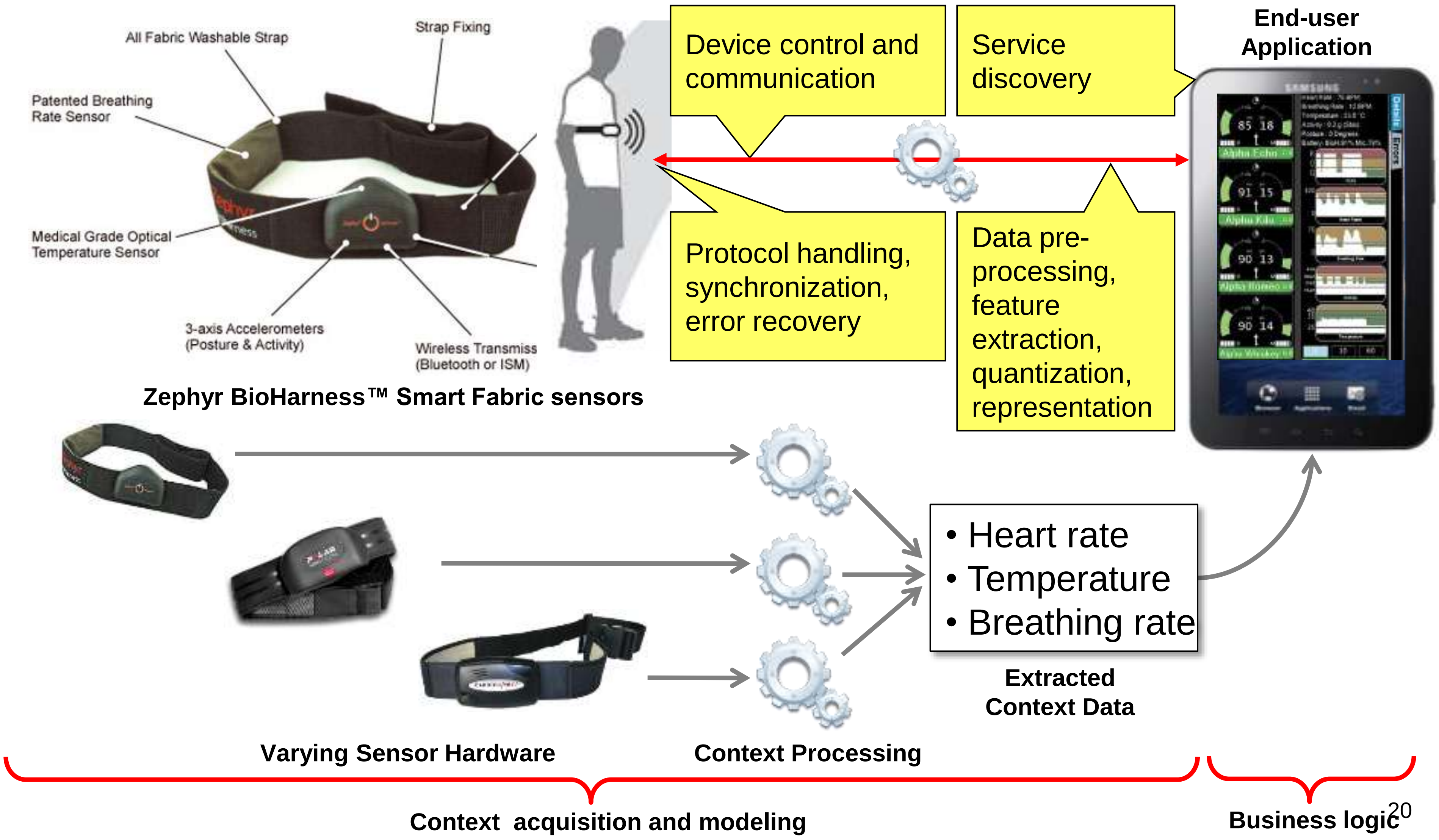
Pedometer Context Plugin ✓
A push-based context plug-in that provides step and associated force information.

Find Plug-ins | Install Plug-ins

Plug-in installation

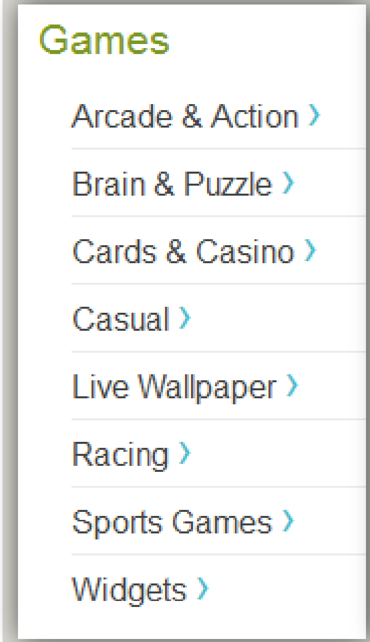
Most of the time, Dynamix is invisible to the end-user

Context Domain Complexity Example (Biotelemetry)


All Fabric Washable Strap
Strap Fixing
Patented Breathing Rate Sensor
Medical Grade Optical Temperature Sensor
3-axis Accelerometers (Posture & Activity)
Wireless Transmiss (Bluetooth or ISM)
Zephyr BioHarness™ Smart Fabric sensors
Device control and communication
Service discovery
End-user Application
Protocol handling, synchronization, error recovery
Data pre-processing, feature extraction, quantization, representation
• Heart rate
• Temperature
• Breathing rate
Extracted Context Data
Varying Sensor Hardware
Context Processing
Context acquisition and modeling
Business logic

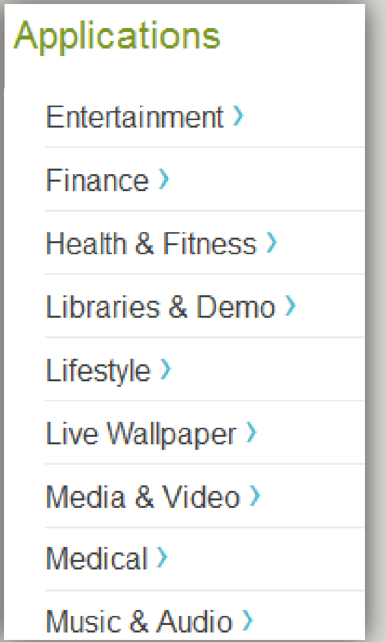
Android Integration

- Dynamix operates as a **service** within Android.
- Developers create Dynamix apps using **existing skills and tooling**.



Games

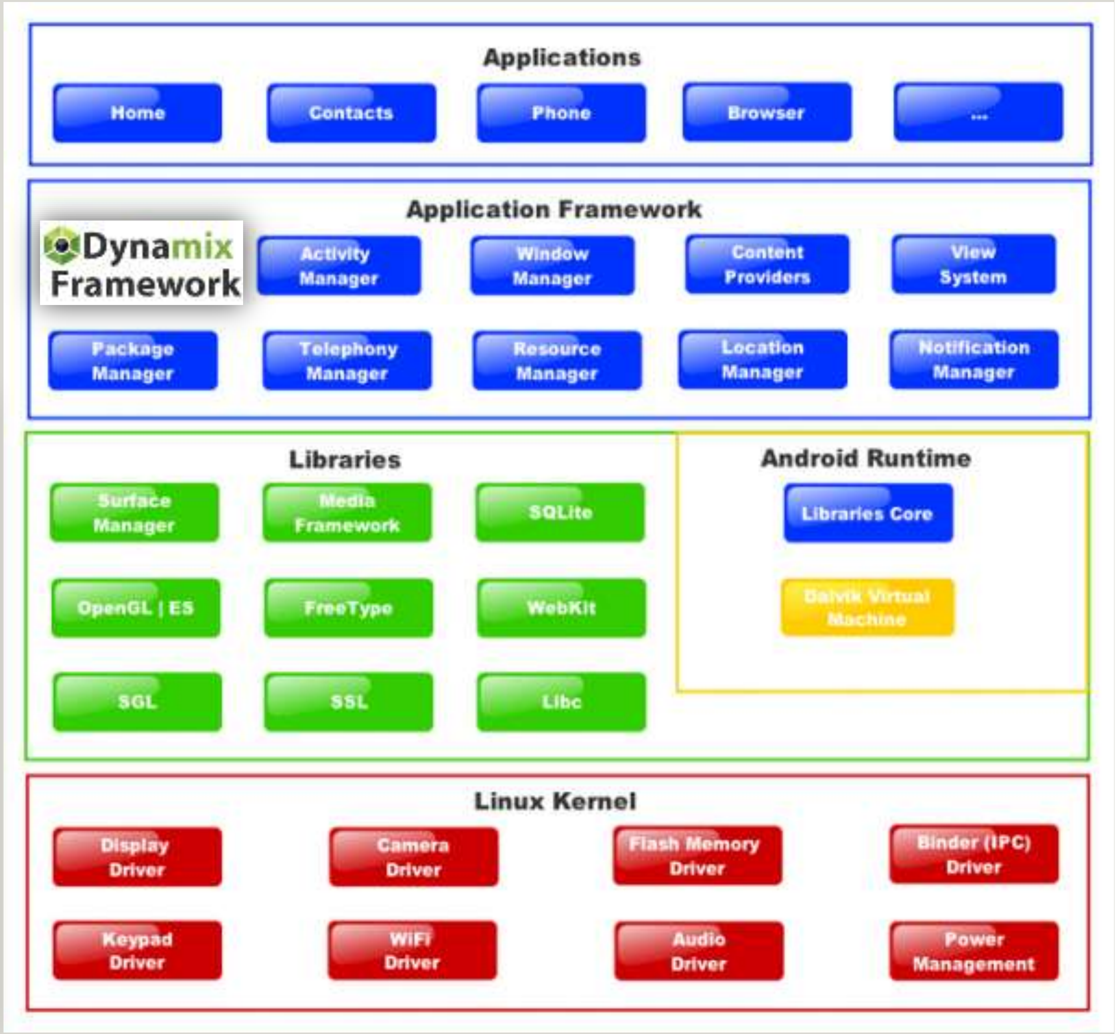
- Arcade & Action ›
- Brain & Puzzle ›
- Cards & Casino ›
- Casual ›
- Live Wallpaper ›
- Racing ›
- Sports Games ›
- Widgets ›



Applications

- Entertainment ›
- Finance ›
- Health & Fitness ›
- Libraries & Demo ›
- Lifestyle ›
- Live Wallpaper ›
- Media & Video ›
- Medical ›
- Music & Audio ›

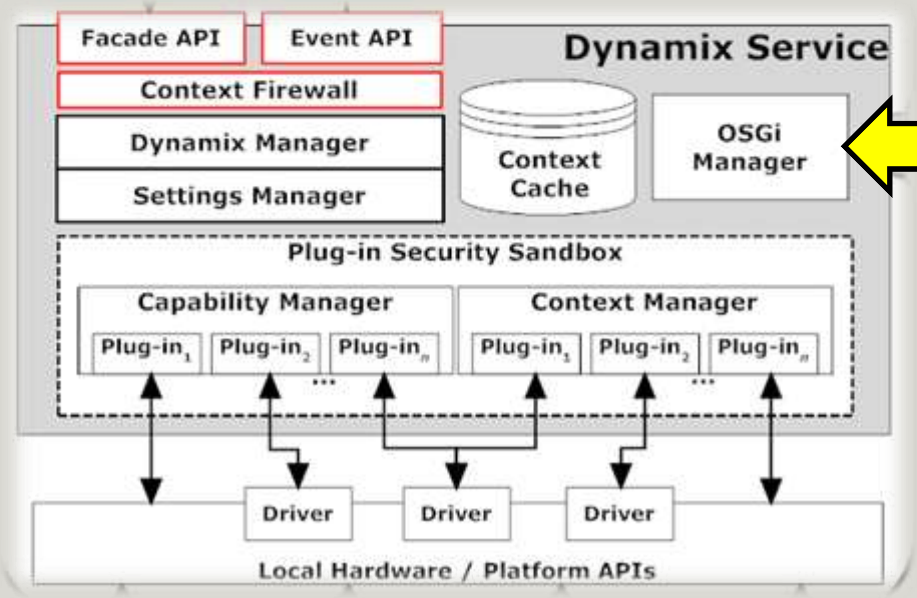
Example app types

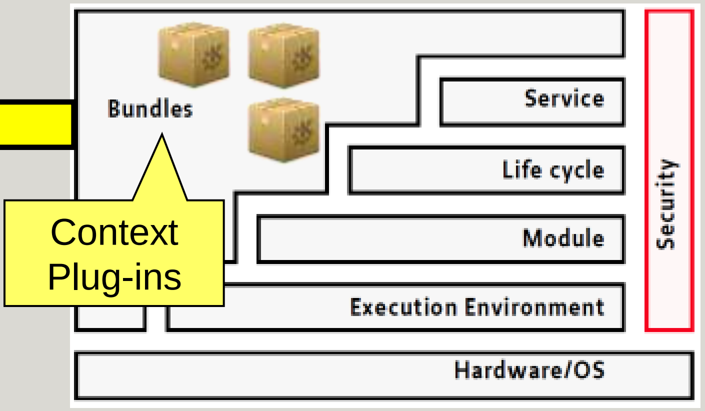


The Android platform stack

Dynamix and OSGi

- Dynamix uses an ***embedded* OSGi Framework** as the foundation of its plug-in architecture (Apache Felix).
- Context plug-ins are packaged and **deployed as OSGi Bundles**.
- The **Dynamix OSGIManager** supports multi-threaded Bundle installations and updates; progress notifications; Bundle verification; runtime integration; and plug-in lifecycle management.

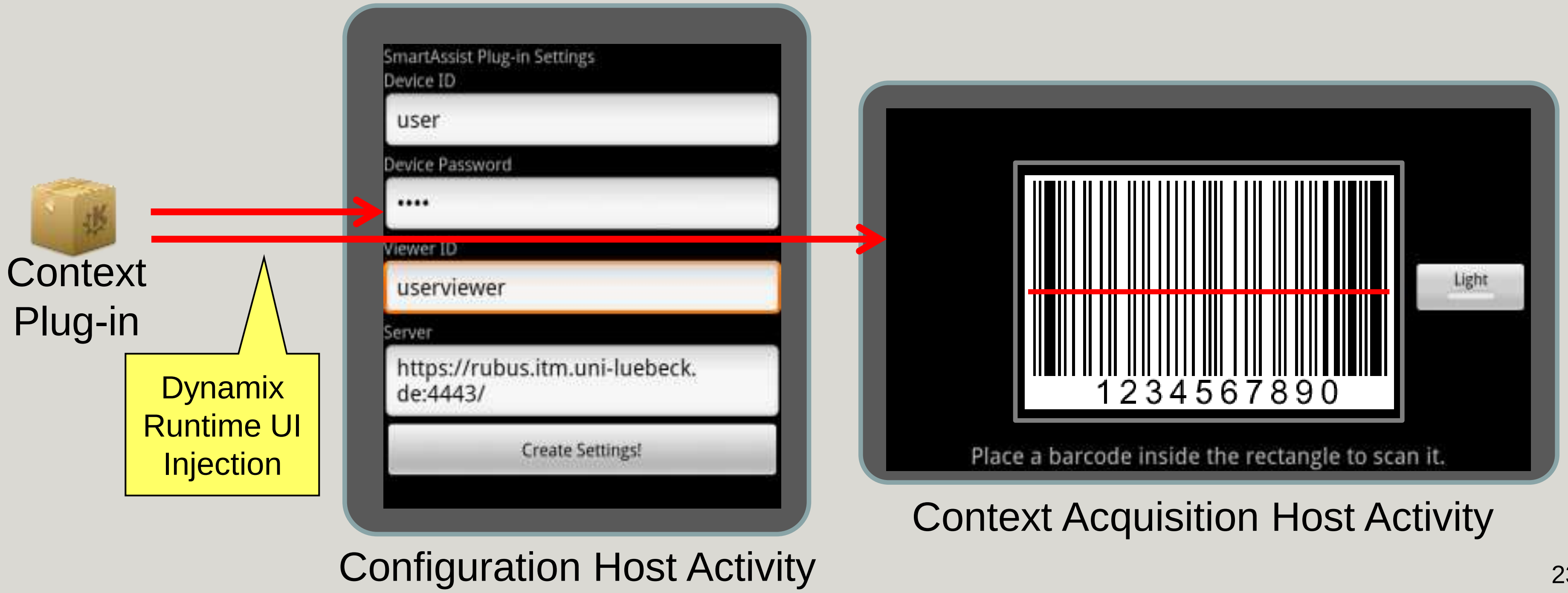




The OSGi Framework Architecture

How Context Plug-ins Interact with the User

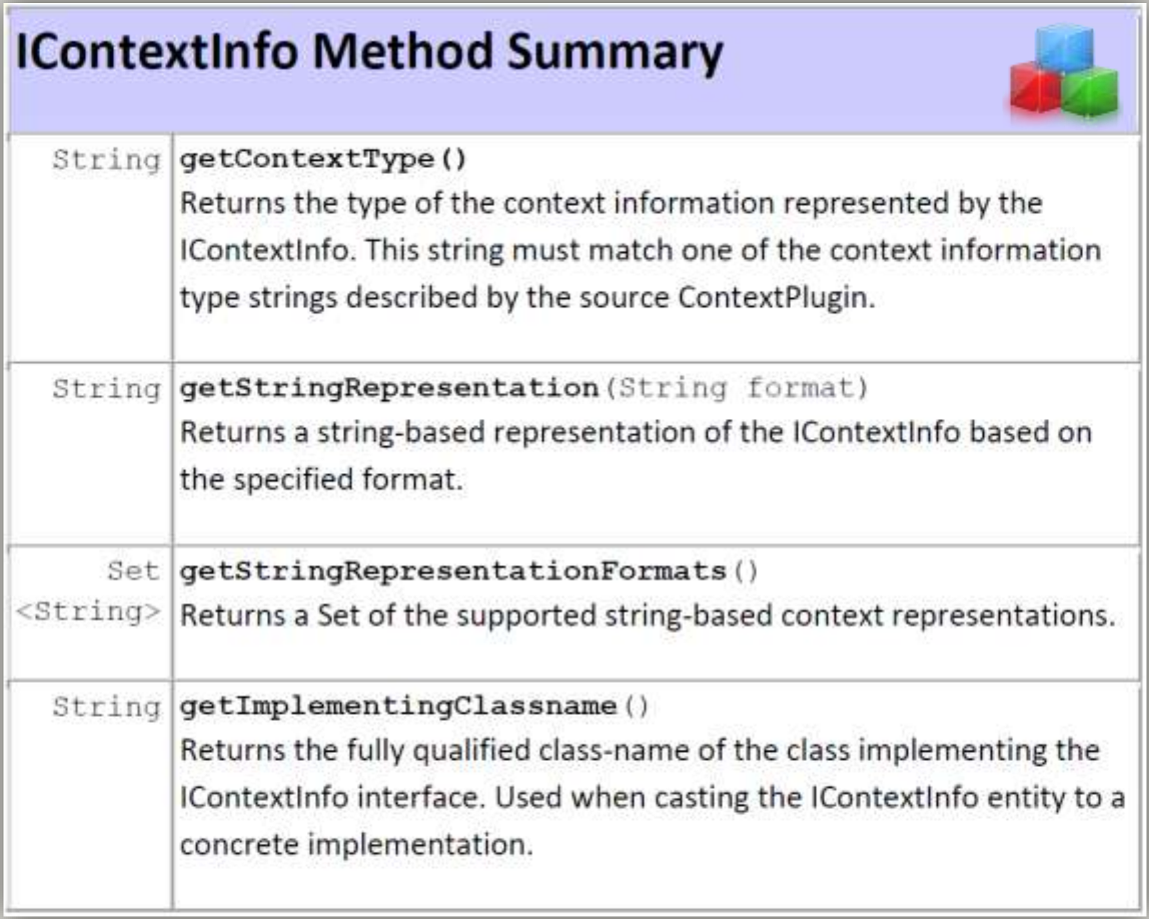
- Some plug-ins may need to provide user interfaces for configuration or context acquisition (e.g. entering data, pointing a camera, etc.)
- However, Android’s security model requires *preregistration* of Activities
- To overcome this problem, Dynamix provides “Host Activities” that can be dynamically populated with a plug-in’s user interfaces.


Context Plug-in
Dynamix Runtime UI Injection
Create Settings!
Light
Configuration Host Activity
Context Acquisition Host Activity

SmartAssist Plug-in Settings
Device ID: user
Device Password: ••••
Viewer ID: userviewer
Server: https://rubus.itm.uni-luebeck.de:4443/

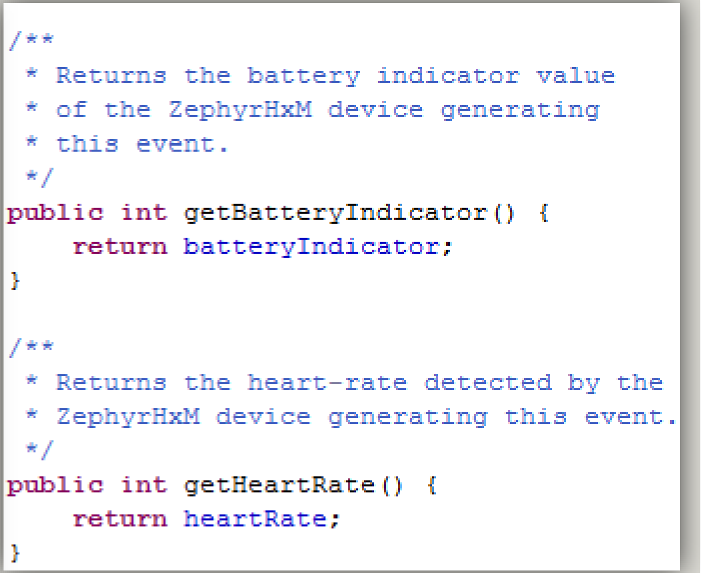
Place a barcode inside the rectangle to scan it.

Representing Context Information with IContextInfo



IContextInfo Method Summary

`String`	`getContextType()` Returns the type of the context information represented by the IContextInfo. This string must match one of the context information type strings described by the source ContextPlugin.
`String`	`getStringRepresentation(String format)` Returns a string-based representation of the IContextInfo based on the specified format.
`Set<String>`	`getStringRepresentationFormats()` Returns a Set of the supported string-based context representations.
`String`	`getImplementingClassname()` Returns the fully qualified class-name of the class implementing the IContextInfo interface. Used when casting the IContextInfo entity to a concrete implementation.



```
/**
 * Returns the battery indicator value
 * of the ZephyrHxM device generating
 * this event.
 */
public int getBatteryIndicator() {
    return batteryIndicator;
}

/**
 * Returns the heart-rate detected by the
 * ZephyrHxM device generating this event.
 */
public int getHeartRate() {
    return heartRate;
}
```

Example IContextInfo code snippet from a heart-rate monitor plug-in

Developers release custom data-types as a standard JAR file, which are used by app developers.