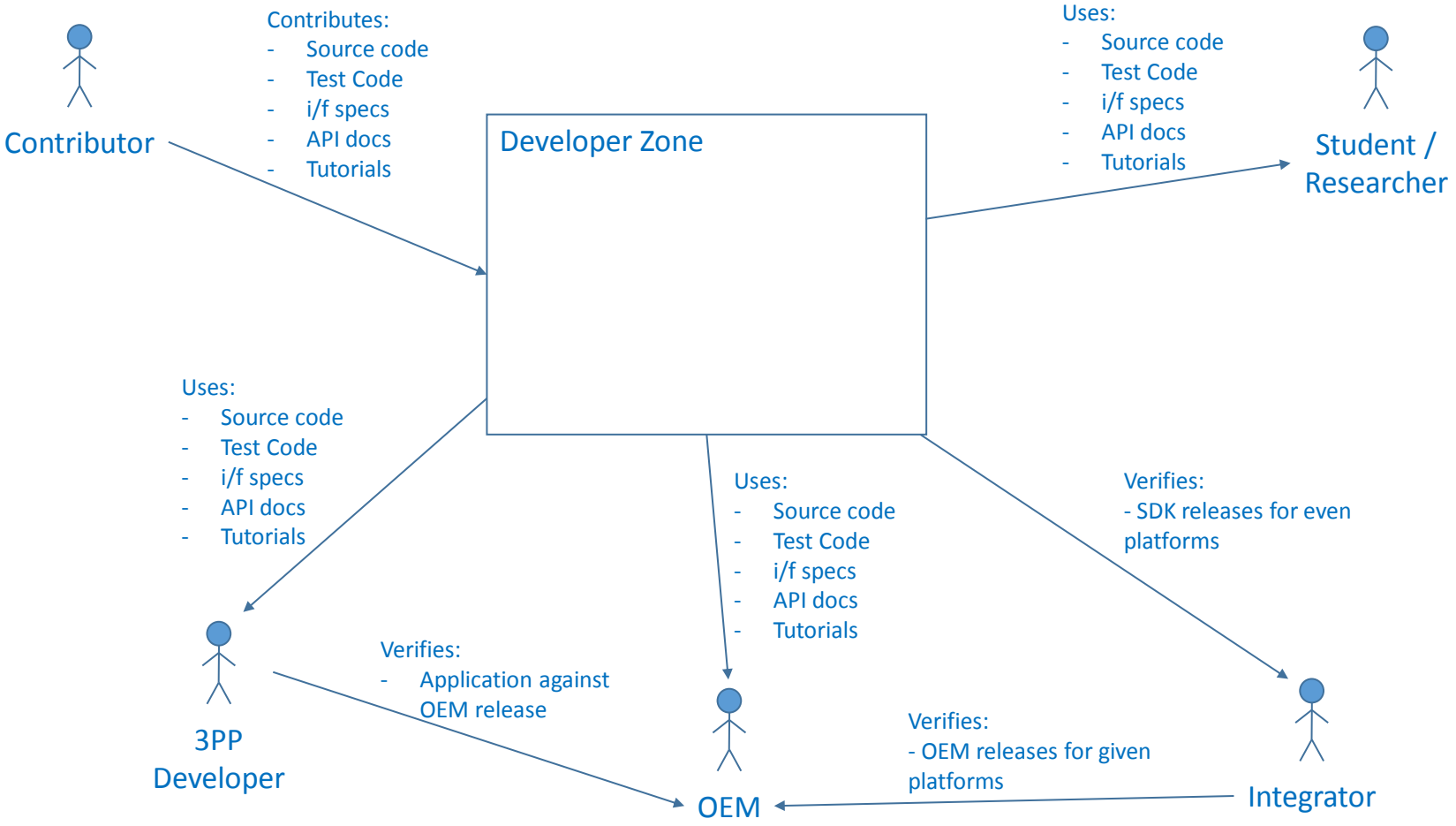
Automotive Grade Android

Unleash the power of *open innovation* for Automotive

Why AGA?

- …an ecosystem (use, share, contribute)
- Enables 3rd party developers to read and write automotive signals.
- Enables OEM (vehicle manufacturers) to build their own Android distribution enabling Infotainment applications to integrate with their vehicles.
- Enables Infotainment applications to combine vehicle data and Internet Services / APIs
- Opens up for new Business Cases/Opportunities

AGA Community Roles



Ways of Working

- Distributed Scrum team at four geographical sites
- 2-3 weeks Sprints (short and intensive)
- Product Backlog (output from pre-study / feasibility) ➔ Features
- Refinements meetings (detailing tasks from the Product Backlog) ➔ Tasks
- Task board
- Daily standups
  - What did I accomplish yesterday?
  - What will I do today?
  - What obstacles are impeding my progress?
  - What can I do to help the team?
- Open Sprint Demonstrations (we invite peoples / parties)
- Retrospective meetings (time for reflections, pros and cons, what can we impr

- Redmine is used for Product Backlog & Task board.

Agile product ownership: https://www.youtube.com/watch?v=502ILHjX9EE

Manifesto for Agile Software Development: http://agilemanifesto.org

User Story: http://en.wikipedia.org/wiki/User_story

Product Backlog
Story 1 *"As a <role>, I want <goal/desire> so that <benefit>"* *"As a <role>, I want <goal/desire>"*
Story 2 *"In order to <receive benefit> as a <role>, I want <goal/desire>"* *"As <who> <when> <where>, I <what> because <why>."*

Story	Task(s)	In Progress	To Verify	Done
Story 1	Task 1-1 Task 1-2 Task 1-3			
Story 2	Task 2-1 Task 2-2 Task 2-2			

"The old way" Waterfall

- Pre-study → Requirement Specification
- Feasibility → Implementation Proposal(s)
- Execution Phase → Implementation & Test & Verification
- Conclusion → Reflections & Final Report

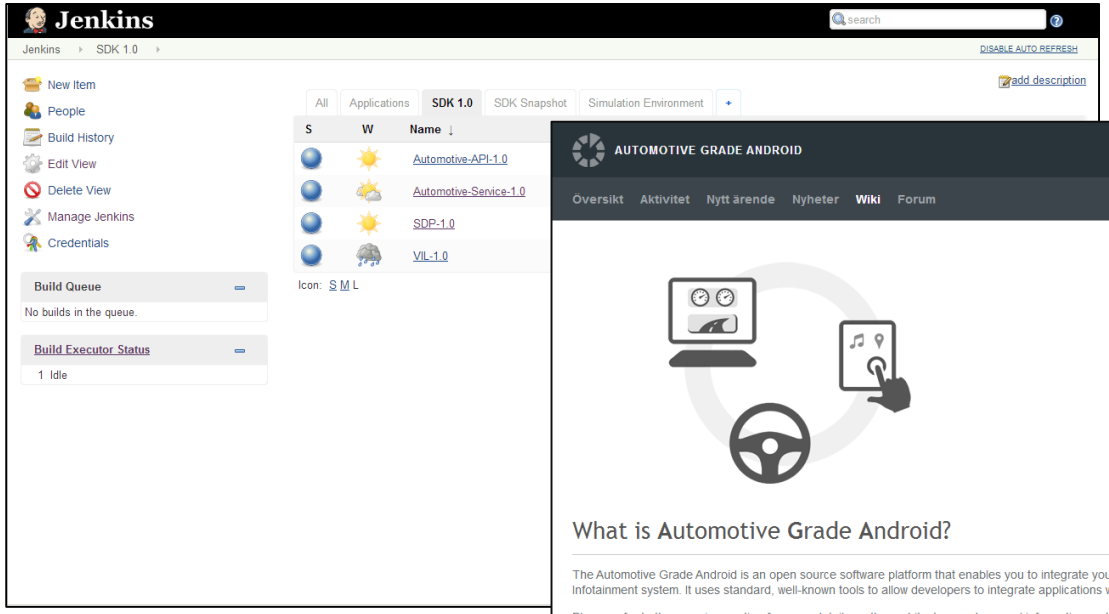
Quite similar to Agile methodologies, though a traditional project may run for months/years. Sprints runs for a few weeks.

Ways of Working cont'd

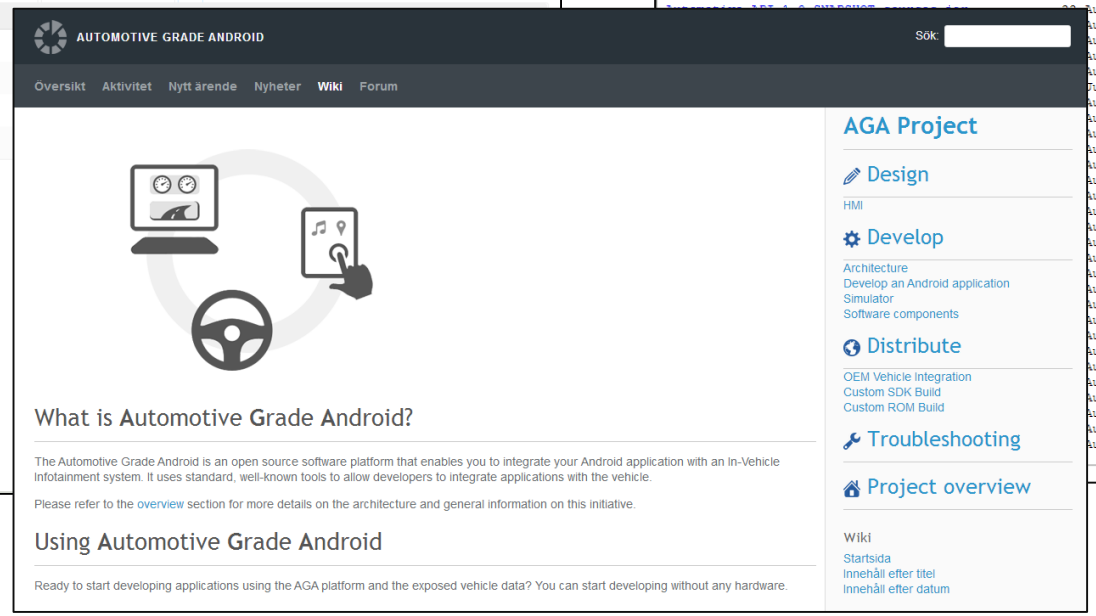
- Tools
  - Google Hangouts (for conference, screen sharing and chat)
  - Redmine (product backlog / task board / wiki)
  - Jenkins (Continuous Integration)

  - Java
  - Git (version control system)
  - Gitolite (access control layer on top of Git)
  - Gradle (build automation)
  - PMD (source code analyzer)
  - JaCoCo (Java Code Coverage)

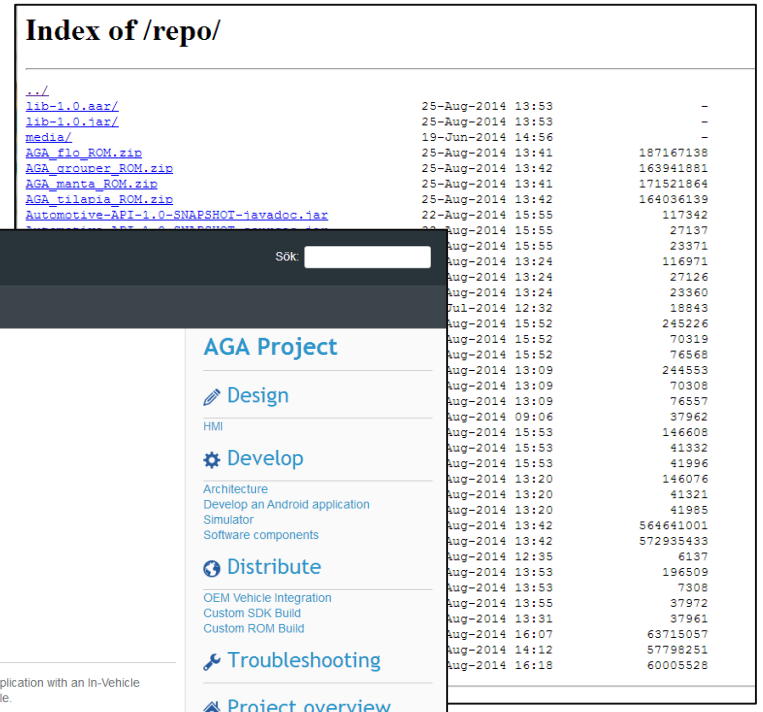
Ways of Working cont'd



Continuous Integration



Wiki

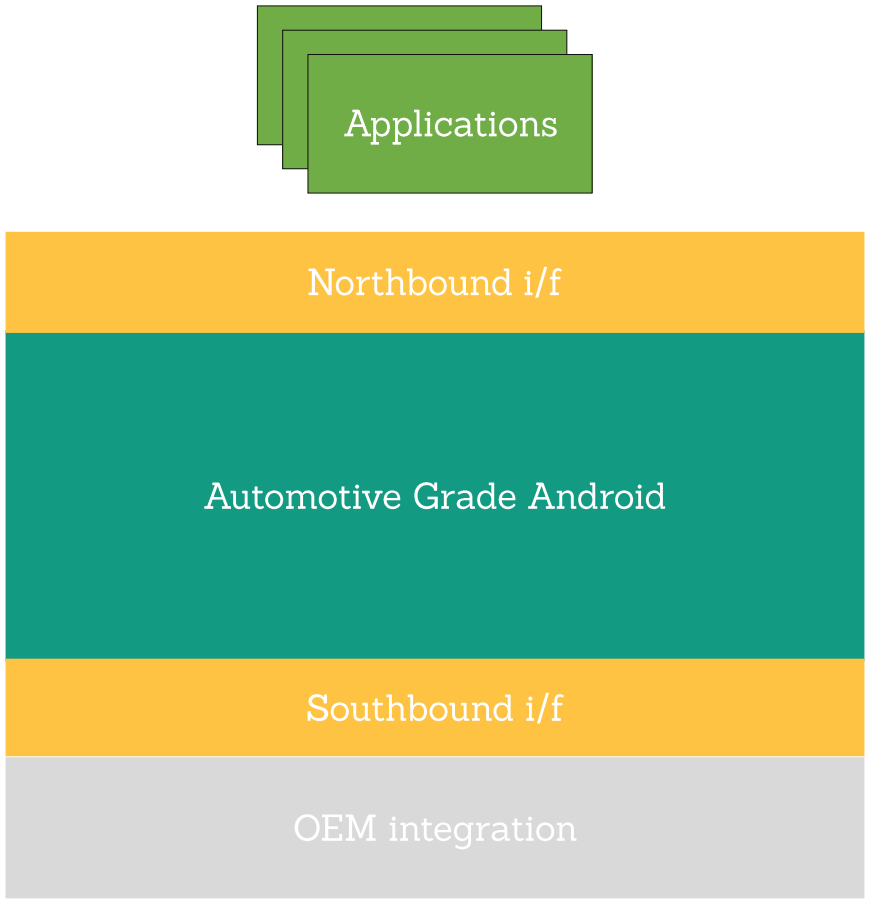


Repo

What AGA Provides & Enables

- SDK
- ROM
- Simulator
- Documentation
- Vehicle signals
- Hardware buttons
- Policy management
- Driver distraction

Logical Overview

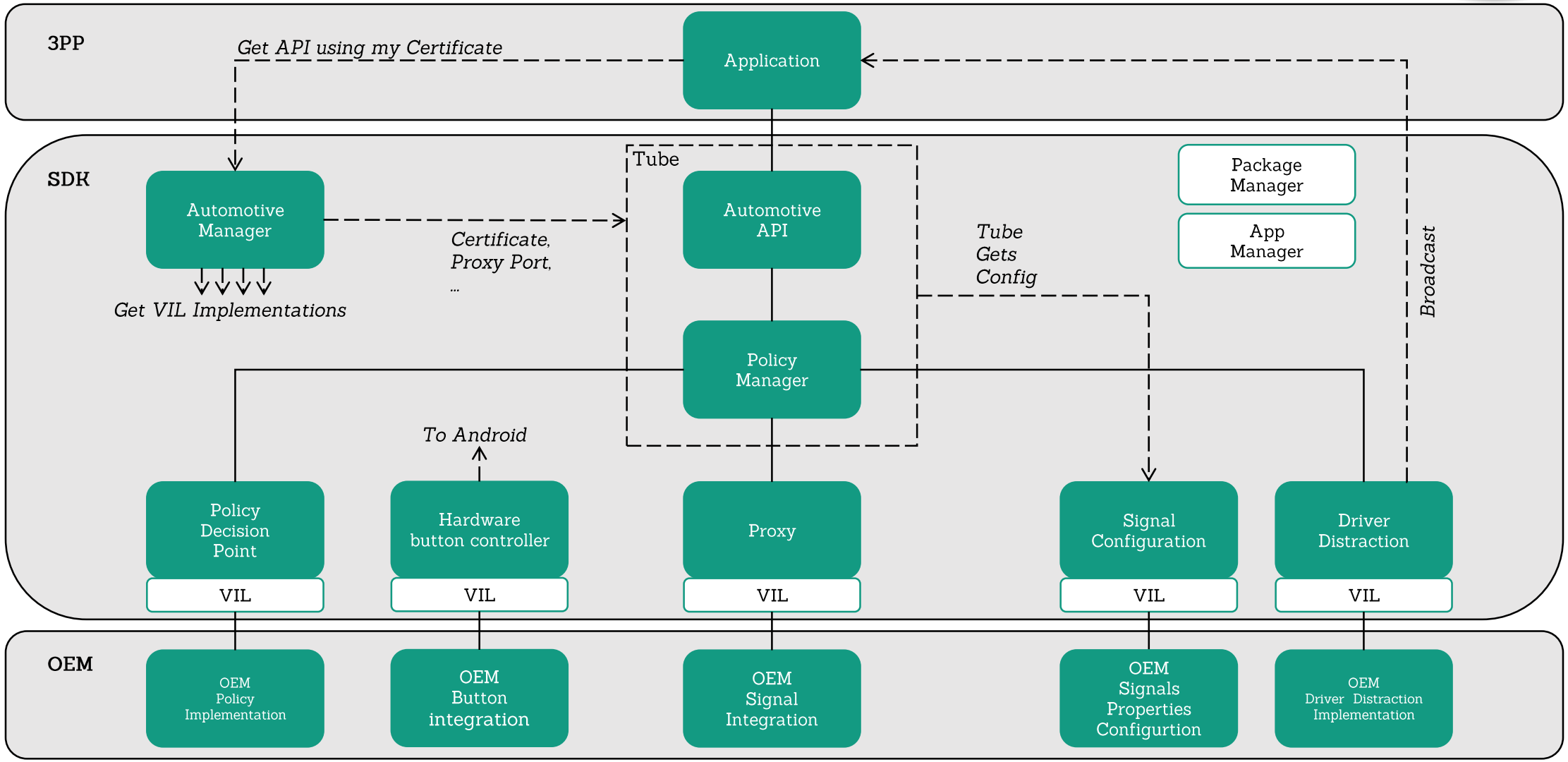


Northbound interface
- Read from vehicle
- Write to vehicle
- React on driver distraction changes

SDK
- Build ROMs
- Emulator
- Simulator
- Documentation

Southbound interface
- Integrate vehicle data
- Integrate hardware buttons
- Set access policies
- Change driver distraction level

AGA Architecture



Vehicle display protocol

Nomadic devices

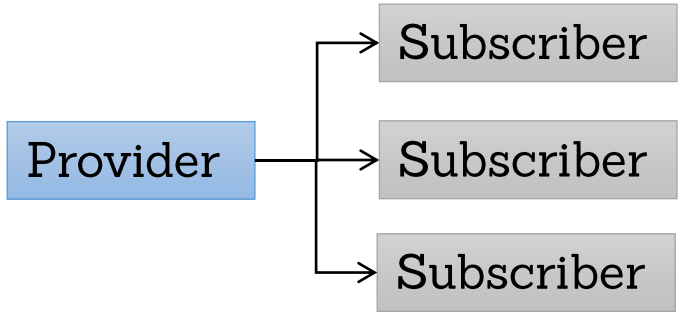
Cluster

Infotainment

SCS / SDP

SDP (System Data Protocol)

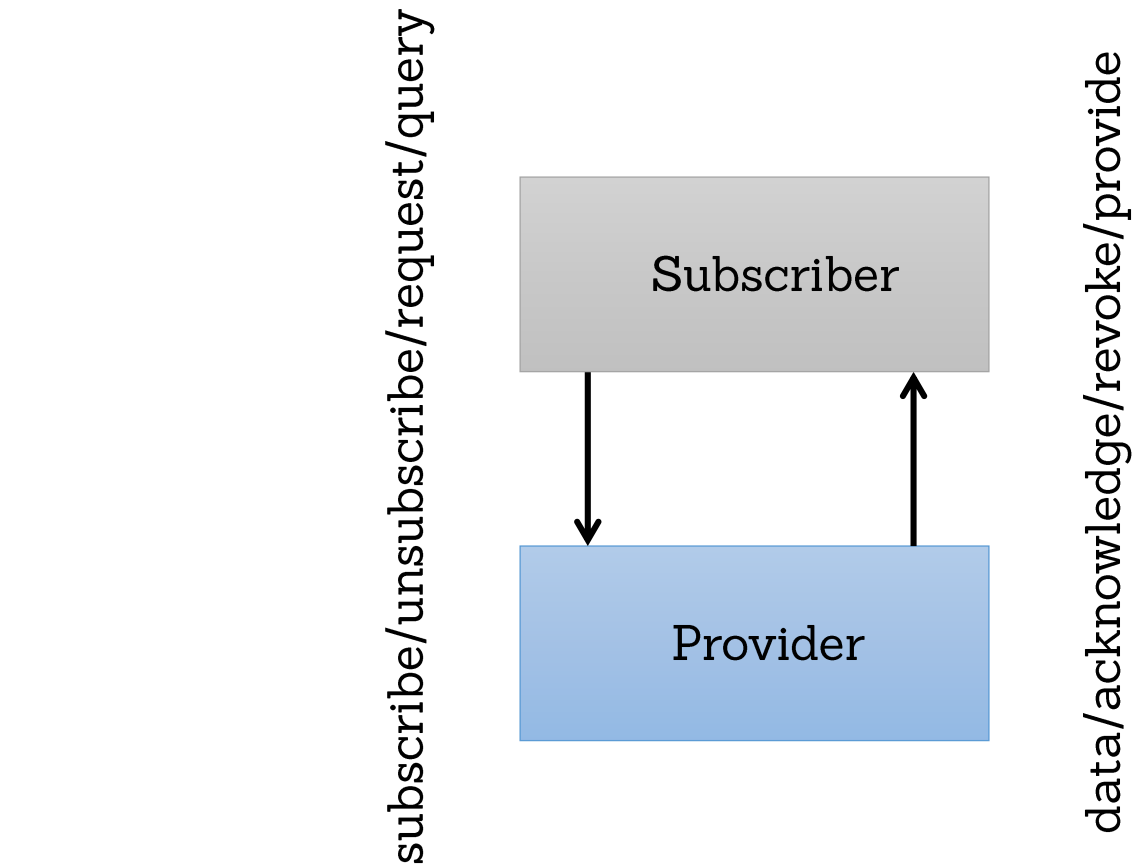
- Make CAN data dynamic
- Provider/Subscriber pattern
- One provider per signal



https://developer.lindholmen.se/redmine/projects/aga/wiki/Software_components

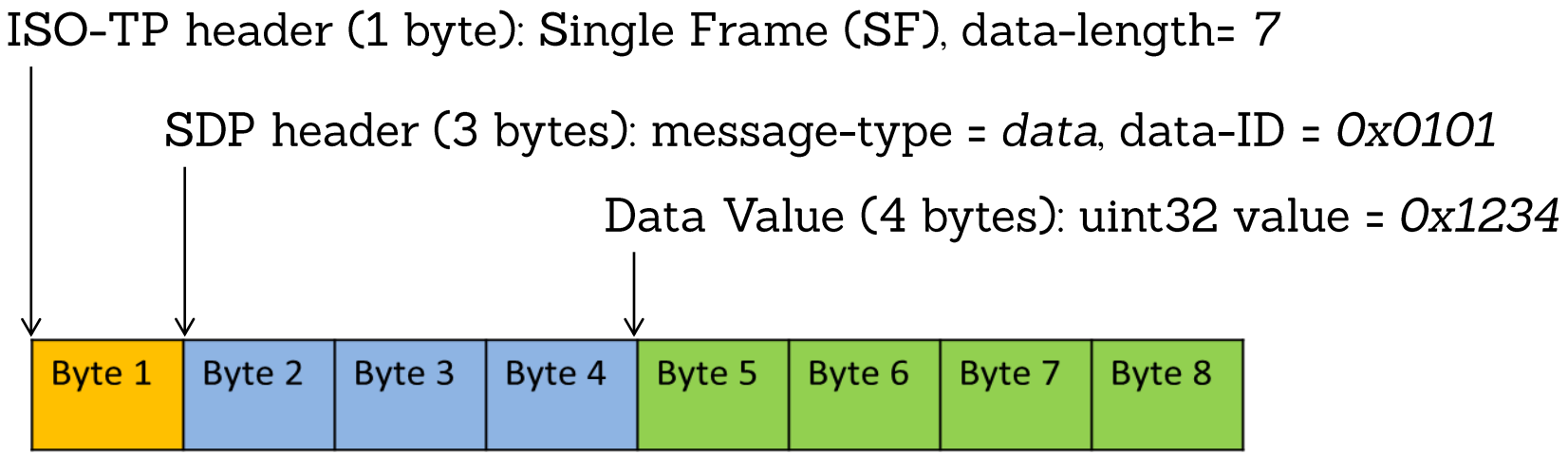
SDP cont'd

- Provide
- Revoke
- Subscribe
- Unsubscribe
- Acknowledge
- Query
- Request
- Data



SDP cont'd

- CAN optimized and light weight
- Signals are addressed with 16 bits
- Up to 32 bits payload in one CAN frame



SCS (Seamless Communication System)

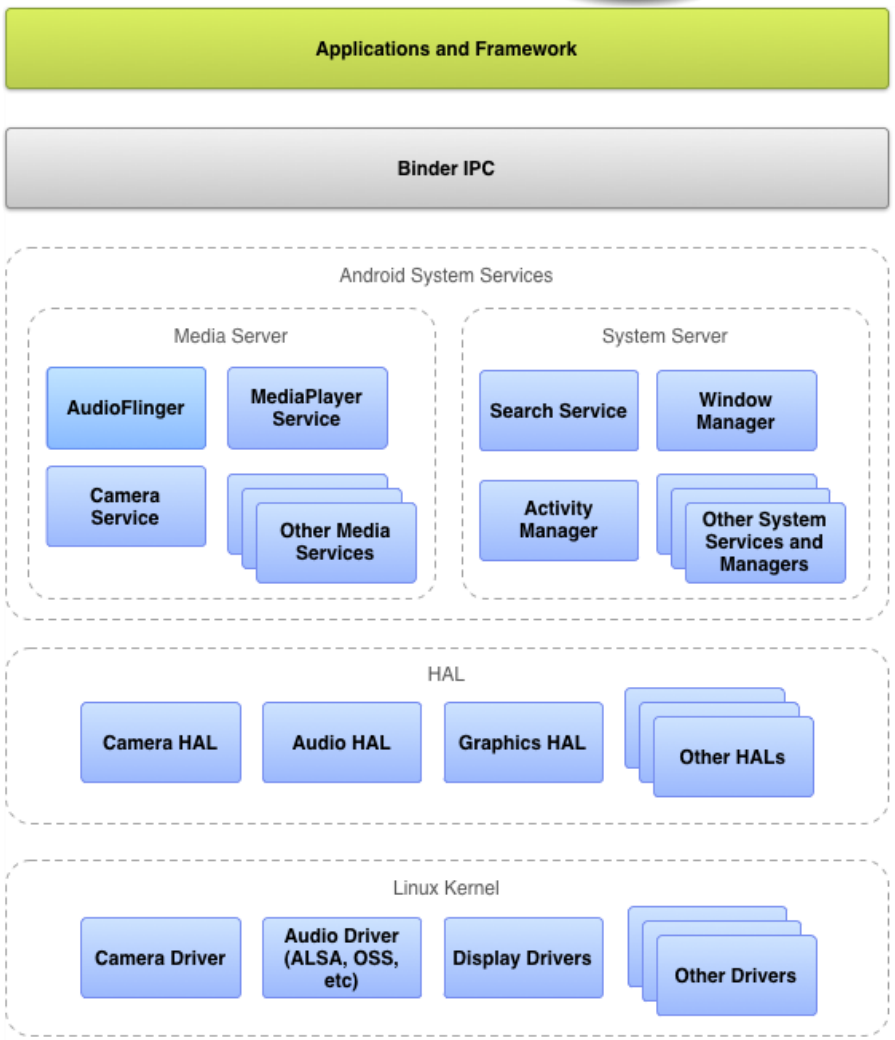
- Configuration of signals
- Payload types
- Can be dynamically changed

- Open source implementation of SDP/SCS is available in AGA

Walkthrough of AGA API

JAR-files vs ROM

- Core functionality written in Java
- Why JARs
  - Easy to develop
  - Easy to test
  - Shorter test cycles
- Cons using JAR-files
  - No Automotive Service
  - Android process split (System vs App context)
  - Only one app at a time (No system proxy)
  - No integrated hardware buttons



Using the API…

Using an AGA ROM

```
final AutomotiveManager manager = (AutomotiveManager)
getApplicationContext().getSystemService(Context.AUTOMOTIVE_
SERVICE);
```

Using JAR files

```
final AutomotiveManager manager =
AutomotiveFactory.createAutomotiveManagerInstance(automotivec
ertificate, automotiveListener, driverDistractionListener);
```

AutomotiveManager setListener

Enables callbacks on signals changes

The listener callbacks

- void receive(AutomotiveSignal signal)
- void timeout(int signalId)
- void notAllowed(int signalId)

AutomotiveManager register/unregister

- register(int... signalIds)
  - Subscribe on data change
  - Data is received in an automotive listener
  - Signals has unique 16 bit id

- unregister(int... signalIds)
  - Unsubscribe signal changes

AutomotiveManager requestValue(int... signalIds)

- Asynchronous call
- Get current value of signal
- One time only

AutomotiveManager send(AutomotiveSignal)

- Application can be provider of signal data

- Application needs the right permissions

Examples

  - Create an application for vehicle climate control
  - Send media information to cluster

class AutomotiveSignalId


```
public static final int FMS_WHEEL_BASED_SPEED = 0x0140;
```

- Configuration

```
CONFIG_INFO.put(0x0140,
    new ContinuosSignalInfo("Wheel based speed",
        "Speed of the vehicle as calculated from wheel or tailshaft speed.",
        100, "KILOMETERS_PER_HOUR", DataType.FLOAT, 0, 300, (1.0 / 256.0)));
```

- Signal ids are 16 bits long
- OEM specific ranges can be added

https://developer.lindholmen.se/redmine/projects/aga/wiki/Signal_IDs

AutomotiveManager isSignalAvailable(int signalId)

- All signals are not available in all vehicles

- Query system to see if signal is configured

AutomotiveManager getSignalInformation(int signalId)

- Get information about signal

- Signal information contains
  - Name
  - Description
  - isDiscrete()
  - Data type
  - Min value
  - Max value
  - Unit
  - Resolution
  - Allowed values
  - Repetition rate

class AutomotiveSignal

- Contains data
  - Signal id
  - Data
  - Unit
  - Time stamp

- Immutable object

interface SCSData

- Blob (byte[]), (i.e. Big Large Object)
- SCSBoolean
- SCSDate
- SCSDouble
- SCSFloat
- SCSImage
- SCSInteger
- SCSLong
- SCSShort
- SCSString
- Uint8
- Uint16
- Uint32

Driver distraction

Using an AGA ROM

```
final BroadcastReceiver receiver = new BroadcastReceiver() {
  @Override
  public void onReceive(Context ctx, Intent intent) {
    final int level = intent.getIntExtra(AutomotiveBroadcast.EXTRA_DRIVER_DISTRACTION_LEVEL, 5);
    Log.i(TAG, ”New driver distraction level: ” + level);
  }
}
final IntentFilter intentFilter = new IntentFilter(AutomotiveBroadcast.ACTION_DRIVER_DISTRACTION_LEVEL_CHANGED);
final Intent currentValue = getApplicationContext.registerReceiver(receiver, intentFilter);
```

Using JAR files

```
final AutomotiveManager manager =
AutomotiveFactory.createAutomotiveManagerInstance(automotivec
ertificate, automotiveListener, driverDistractionListener);
```

Hands on

System.out.println("Let us get practical with AGA!");

Questions

Contacts

Magnus Nylén
magnus.nylen@swedspot.com

Joakim Lundvall
joakim.lundvall@swedspot.com