

# OSGi Automotive Workshop



March 22, 2007  
Siemens VDO - Eindhoven

# Agenda

Moderator: Peter Kriens, Technical Director & Evangelist,  
OSGi-Alliance

- 08.30 Arrival and coffee
- 09.00 Welcome
- 09.15 Introduction OSGi and Vehicle Expert Group
- 09.45 Participants Presentations
  - Workshop participants can give a brief presentation (5 min) on desired areas of work
- 10.30 Results of the Automotive Workshop in Troy
- 11.00 Open Discussion
  - brainstorming on topics that VEG should work on
  - Look for common interest of areas
- 12.00 Prioritizing ideas
- 12.30 Lunch
- 13.30 Continued open discussion based on priority
- 15.30 OSGi organization, technical process, IPR overview, next steps
- 16.30 Adjourn

# OSGi Alliance

- Fosters an eco-system of hardware manufacturers, infrastructure providers, software developers, and service providers with the goal to create a common market for networked software services
  - Manages a solid specification process to create a standardized service platform that is usable across industries and hardware
  - Manages a certification process to guarantee interoperability
- Driven by real requirements from different markets

# Convergence Is Key



Automotive



Communication



Mobile To Mobile

Working People



# OSGi Technology

- OSGi technology is *Universal Middleware*.
  - OSGi technology provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle.
- OSGi technology is
  - Cross industry (mobile, vehicle, communications, administrative, software development)
  - Cross platform (embedded, PC, server, mainframe)

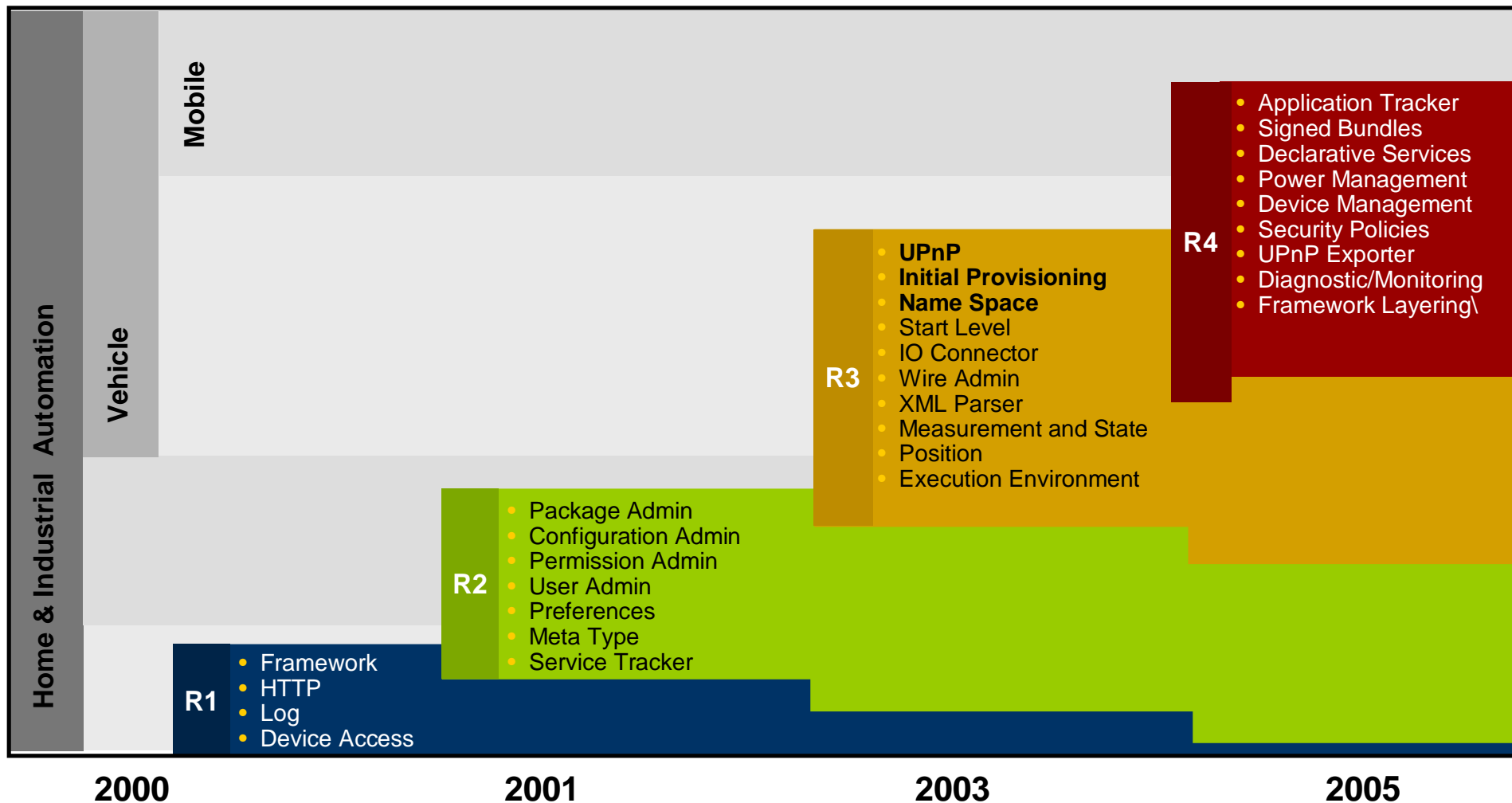
# OSGi Alliance

- Specifications are driven by members
  - Members write RFCs
  - Expert Groups review RFCs in teleconferences and meetings
  - RFCs are turned into specifications by a professional specification editor
- The OSGi Alliance can not “surge” the troops. High quality specification writing is hard work for members!

# Vehicle Expert Group

- Founded in 2002
- Charter
  - ... According to the OSGi policy, the VEG needs input from automotive, transport and telematics companies, as well as from other OSGi groups and other standardization organizations.
- Key players Smartmove/ACUNIA, BMW, Siemens VDO, ProSyst, Gatespace Telematics
- Related Projects and Products
  - Ertico 3GT
  - AMI-C
  - Ertico GST
  - BMW 5 series

# OSGi Service Platform



# VEG Pipeline

- Vehicle API
  - Provides access to low level sensors, actuators, and state of the vehicle
  - Based on OMA DM, a standard mobile device management protocol
- Diagnostics
  - Provides the possibility to download statistics and run diagnostic tests
- Power Management
  - Provides applications with the possibility to adapt their performance to the power state of the vehicle.
- Navigation API
  - Provides applications with the possibility to interact with the navigation system. Pretty cool!

# Why OSGi Technology?

- Because there should be a better way to develop software ...



# Participant Presentations



# About the Workshop



# Workshop Goals

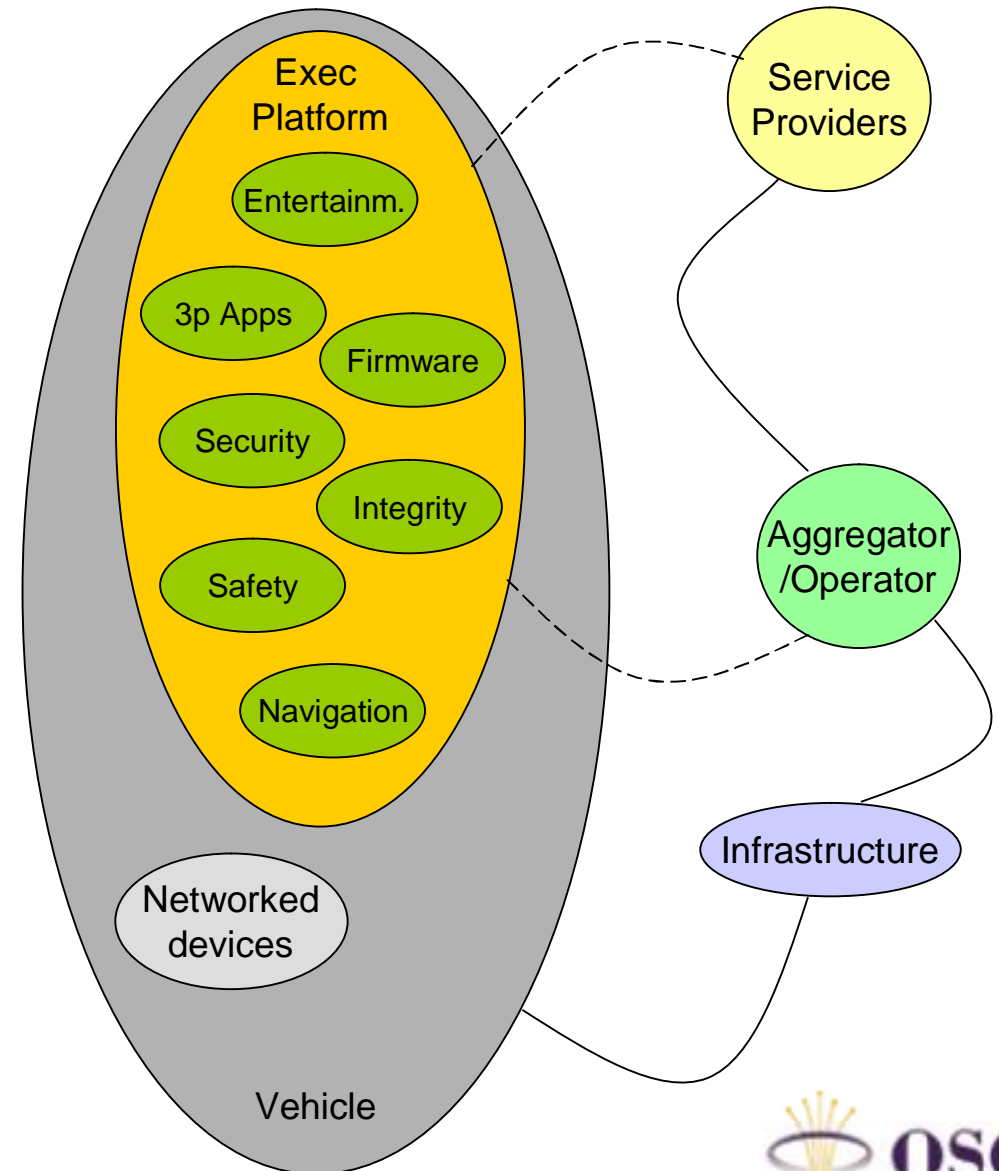
- Organize the multiple ad-hoc efforts in applying OSGi technology in vehicles
  - Ertico 3GT and GST projects
  - Ami-C
  - VII
  - ...
- Identify necessary liaisons
- Create a prioritized list of RFCs to make
- Sign up for work!

# Question 1: VEG Goal

- Create third party bundle market for applications running in vehicles?
  - Entertainment?
  - Enterprise?
  - Government mandated?
- Simplify software development for the OEMs by providing a common platform to their suppliers?
- Create a business model for service providers, aggregators, and OEMs based on networked services?
- Provide a technical foundation for projects like VII, GST, and road pricing?

# Question 2: What is the scope?

- What should the VEG scope be?
  - The OSGi provides an execution environment, what kind of applications should run on it?
    - Entertainment, Third party software (open/closed), Firmware (functions required by the car to operate), Security, Integrity, Safety, Navigation, ...
  - What should it integrate with?
    - Devices on the can/most/... busses?
    - To what level, should it handle the software on these devices?
  - Is the infrastructure included?
    - Should the same middleware run on the vehicle as well as in the net infrastructure?
  - Are business models included?
  - Are standards (protocols and APIs) needed for the Aggregator and Service Providers?



## Question 3: Where is the money?

- Specifications are hard work requiring a significant investment by the participants
  - What are the drivers for the participants to do this hard work?
  - Who will benefit from the work?
- What are the opportunities, what are the threats?

# Open Discussion



# From Troy: List of Areas

- Safety
  - Use case: Stop when car in front of you brakes. Vehicle – Vehicle communication, event notification. Hint to the user, driver stays responsible? Brake activation? What is part of the impl.
  - Use case: Safe following distance
  - Generic Safety oriented service that provides information about the road conditions and other vehicles and the proximity
- Provisioning
  - Use case: OEMs can push bundles to the OBEs
  - Requires also deployment of non-OSGi parts
- Resource allocation policies/guarantees and hard real time issues for bundles
  - ARINC avionics standard in this area or relate to other standards
  - Prioritization, what they mean
  - Scheduling
- Non-Java code integration
  - Includes for example vml and framework, but also other apps.
- Vehicle Interface
  - Location, speed, temperature, heading, windspeed, traction, abs act. Airbag deployment, brake lights, technically any sensor
- HMI
  - Make it easier to integrate with existing HMIs
  - AMI-C already started this
  - VII will have HMI interaction requirements
- Security:
  - Use case: how to secure sessions and how to secure the application integrity
  - How to authenticate the vehicle (uses x509 certs) 1609.2 protocol
  - Confidentiality, integrity, authorization, authentication
  - Management of certificates, upload certificates

# Troy Results

- Privacy
- Reprogramming ECUs
  - Issues with anonymity
  - How does this interact with existing standards around the vehicle databus
- Will OEMs open the OBE open up to 3p apps?
  - Will likely be the case
- Lobby with governments to require OSGi
  - Toll fee collection in Germany
  - In US states make a lot of decisions
- Drivers
  - Companies have to make too many drivers today
  - Standardized drivers
  - To connect anything there is out there
  - Resource and time critical requirements
- AutoSAR
  - Overlap?
  - Could autosar apis be available to VM appl.
  - Where do they connect

# Troy Results

- Diagnostics
  - “offboard”
  - Access to detailed static data : VIN, model, type, options, engine, etc.
  - Access to real time data: engine speed, cooling temp, etc.
  - Abstraction needed because there is so much variation on the bottom
  - Choose network based on speed, acces, etc.
  - Bus types abstractions
  - Multiple levels of security
  - Needs safeguards
  - Run special code on demand, download over the air
- Download data like radio tuning, firmware of ecus, codecs, calibration files, scripts, etc.
- OSGi light
  - Something that can on a smaller device
  - What memory budget
  - What services:
  - Flexibility to create smaller impls. Of OSGi based systems
- Slaved to a core OSGi, with reduced functionality
- Distributed OSGi
- Navigation Model
- Liaison with CVTA
  - Depends on VEG status
- Make OSGi better known to people that decide what goes into the vehicle
  - Provide articles, whitepapers, convergence
  - Get rid of the question: OSG what?

# Prioritizing



# Troy: Priorities

- 7. Resource allocation policies/guarantees and hard real time issues for bundles
- 5. Vehicle Interface
- 5. HMI
- 4. Non-Java code integration
- 4. AutoSAR
- 3. Safety Information
- 3. Provisioning
- 3. Drivers
- 3. Diagnostics
- 3. Download data like radio tuning, firmware of ecus, codecs, calibration files, scr
- 3. Navigation Model
- 2. Security
  - 2. Lobby with governments to require OSGi
  - 2. OSGi light
    - Privacy
    - Reprogramming ECUs
    - Will OEMs open the OBE open up to 3p apps?
    - Slaved to a core OSGi, with reduced functionality
    - Distributed OSGi
    - Liaison with CVTA
    - Make OSGi better known to people that decide what goes into the vehicle

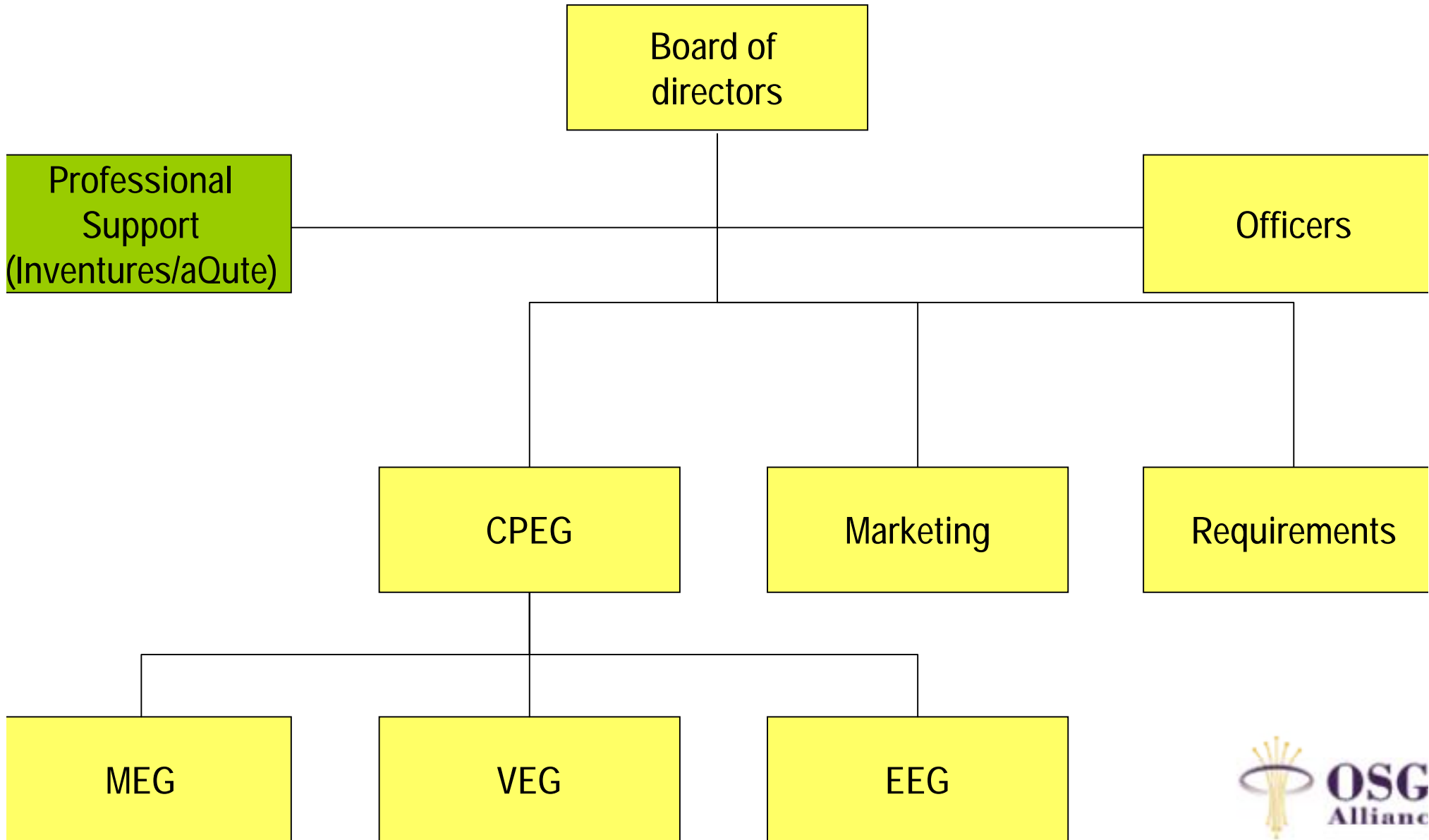
# OSGi Process Info (The exciting part)



# OSGi Alliance Background

- Founded in 1999
- Initial focus home automation/embedded use. Today specifications are used in home, automotive, and mobile applications
- Currently Release 4
- Highly applicable in almost any computing situation
- Key features
  - Modularized
  - Dynamic
  - Secure
  - Manageable
- Current focus on becoming a standardized layer for universal middleware
  - Cross industry
  - Cross platform

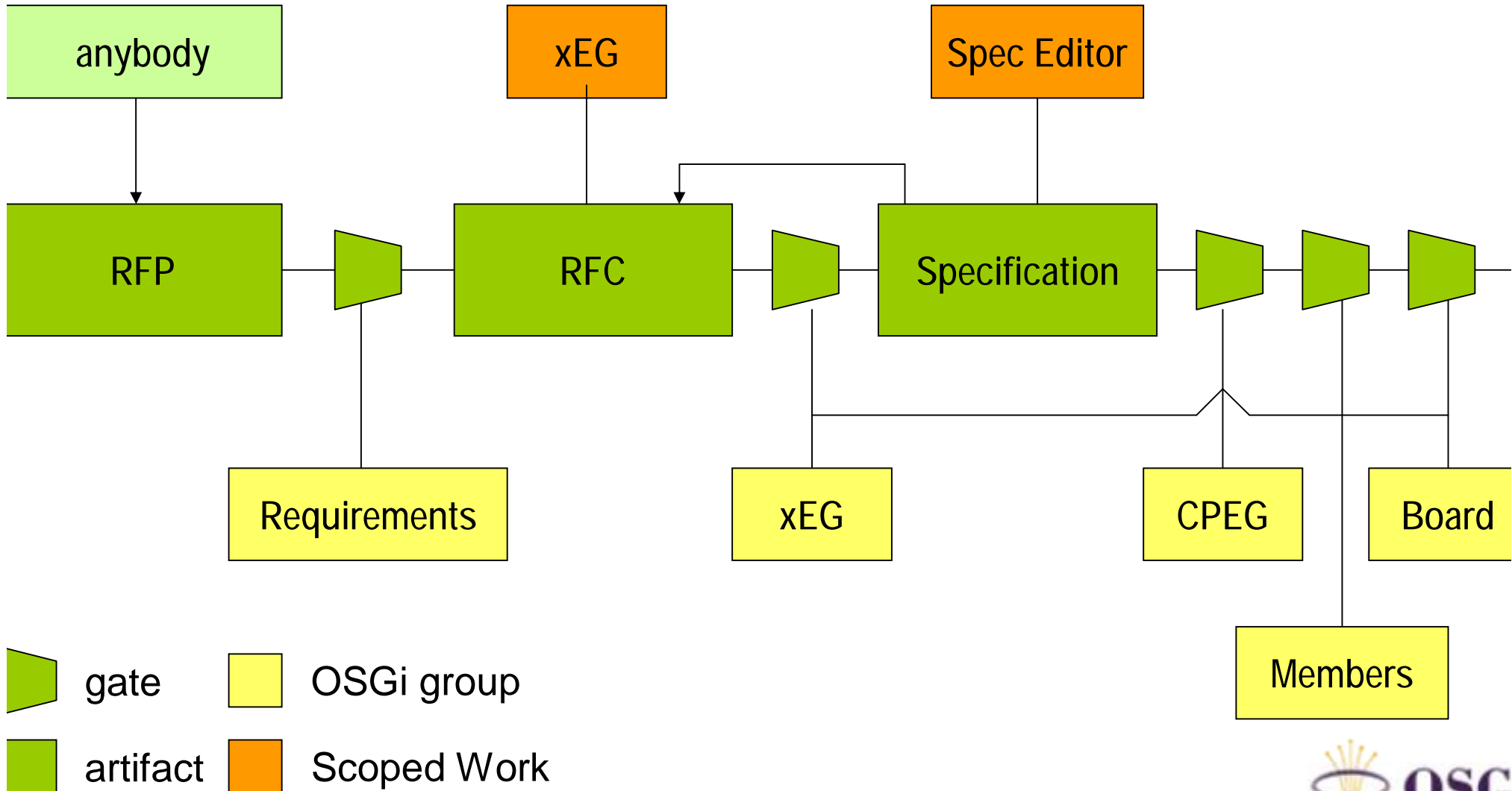
# Organization



# Technical Process

- Experts Groups
  - Charter (scopes the activities of the EG)
  - Statement of Work (SOW) scopes the activities of the members in the EG
- RFP
  - Requirements document can come from external sources
- RFC
  - Internal design documents, developed within an EG
- Specifications
  - Specification documents for external consumption
- Reference Implementations of *all* specifications
  - Can come from open source groups
- Test Compatibility Suites
  - Developed by EG members

# Process



# Next Steps

- Join OSGi as member to participate in the Vehicle Expert Group
- Agree for a kick-off meeting
- Write RFCs
- ...

# Priority List



# List

- 4 Lightweight
- 1 Resource saving
- 3 Performance
- 6 Deterministic handling of resources
- 7 Hard/soft real time
- 5 Forced unload of bundles
- 5 Utility/Resource manager
- 8 Simple integration of C++ components
- 0 Interfaces to standards applications like nav. systems
- 5 Software driver support
- 8 Universal OSGi

- 3 Set Log level per bundle
- 1 Inspect log level of a bundle
- 0 Current OSGi Position object too complex
- 3 Enhance positioning service
- 9 Navigation
- 4 CAN
- 0 Flexray
- 0 Use Vehicle Admin Tree from GST
- 11 Vehicle API
- 7 Integration with AutoSAR
- 1 Inspect internal state of OSGi services
- 0 Interface function to stop or restart OSGi services
- 0 Also remotely

- 3 Bundles can move between frameworks
- 1 Uniform interface for depl. Of services
- 0 JSR 232 Depl. Adm. Spec.
- 3 Possibility to update single services bundles of a depl. Package
- 1 Alter pars. of configuration settings over a remote interface
- 0 OSGi service similar to Preferences Service or Windows registry
- 0 Global access to all services
- 0 Config admin is too complicated
- 6 Simple embedded database or interface to it
- 3 Object relational mapping
- 2 Support for simple TXs
- 4 Data synchronization service
- 9 Central administration point for communications
- 2 Simple selection of appropriate Comm. Channel

- 4 Support ad hoc networks
- 5 Data transfer via web services
- 5 Optimization for low bandwidth and unstable links
- 0 Use model driven development for generation of data object and client stubs
- 4 Secure comm.
- 0 Link software to a defined device
- 1 Signed bundles
- 1 Confidentiality
- 1 Integrity protection
- 1 Introduce security manager
- 2 Secure logs
- 1 Secure persistency
- 0 Is DRM an issue?
- 1 Lock data, configurations to a device

- 5 Flexible HMI
- 3 HMI Updatable during appl. Lifetime
- 6 Provide an API to HMI
- 1 How to interact with a user
- 0 Customizable HMI
- 5 Sharpen the VEG vision
- 3 Define clear VEG goals
- 0 Prioritize
- 0 Which domains should VEG work in
- 0 Many supported HW/OS platforms

- 0 Becoming an industry standard
- 2 Harmonization with JSR/JCP, Eclipse
- 0 Harmonization with current JVM level
- 1 Better tool support could make OSGi more attractive for developers and management
- 0 Config. Bundles settings and other attrs
- 0 Generation of interfaces from UML MDA/MDD
- 2 Debugging
- 0 Cross platform development
- 1 Automotive industry AMI-C use cases

# Priority List from Eindhoven

11	Vehicle API. Uniform access to the static and dynamic vehicle information, read write, cross manufacturer, cross bus. A Vehicle Object mode. There are currently three proposals on the table: GST, OSGi, Volkswagen (based on OSGi). Requires security and authorization.	Kruger & Fr
4	CAN. Same as the above ... This NOT a CAN bus abstraction.	
9	Navigation API. An API for location based services. POI, positioning, routing, guidance, map handling (update/extend the map OTA), traffic information.	Poggeman
3	Enhance positioning service (Position object is just guessed position, need more accurate map matched based, Navigation)	
9	Communication Manager. Enables application to choose a communication link. Central administration point for comm. Channels. Enumerate the channels. Unify different comm. Technologies. Cheapest, fastest, only local hotspot, ..., nearest roadside unit. From an appl. perspective it should simplify access to a comm. link. Should address sharing. Prioritization.	ProSyst
5	Optimization for low bandwidth and unstable links	
4	Support ad hoc networks	
2	Simple selection of appropriate Comm. Channel	
8	Simple integration of C++ components (NOT JNI)	van den Be
8	Universal OSGi. Allows native code to be deployed and managed as bundles, as well as enabling native code to use the service registry. Enable distributing the firmware to other components? AutoSAR link?	
7	Hard/soft real time.	Coates
6	Deterministic handling of resources (enforce, tooling). Offline, ahead of time, runtime	
7	Integration with AutoSAR. How to communicate between an AutoSAR defined system and OSGi based applications. Market force. Can OSGi integrate with the AutoSAR tools?	Mueller
6	Simple embedded database or interface to it. Persistence storage of data, maybe higher level model like hibernate, toplink, JDO, etc. Higher level than JDBC.	Maintz
3	Object relational mapping	
6	Provide an API to HMI. Bindings to different UI technologies. Abstraction for UI events/event handling. Separate logic from the UI, where is the edge thin client. Look at themes and how bundles can integrate into resident themes. Non graphic Uis.	Haak