PTV Project Related OSGi Requirements

**Functional Enhancements**

- **HW/OS resources**
  - Lightweight, resource saving, performance
  - Deterministic handling of resources
  - Hard/soft real time
  - Forced unload of a bundle (bundle terminator)
  - Utility / Resource Manager
  - Simple integration of C++ Components
  - Interfaces to standard applications like

- **Compatibility**
  - Software driver support for e.g.
  - Universal OSGi
  - Log level per bundle
  - Inspect current log level of a given bundle

- **Logservice**
  - Current OSGi position object is too complex and to limited
  - Enhance the Positioning Service towards
  - Navigation simple model vs. complete model

- **Navigation/Positioning**
  - CAN
  - Flexray
  - Use Vehicle Admin Tree from GST?
  - Vehicle API
  - Integration with AutoSAR
  - Diagnostic interfaces for OSGi-Services
    - Inspect internal state of OSGi services
    - Interface function to stop or restart OSGi service from higher level "service manager"
    - Also remotely

- **Distribution**
  - Bundles can move between frameworks

- **Remote Deployment**
  - Uniform interface for deployment of services
  - JSR 232 Deployment Admin Specification
  - Possibility to update single services/bundles of a deployment package

- **Remote Management**
  - Alter parameters, configuration settings over remote interface

- **Registry Service**
  - OSGi service similar to Preferences Service or Windows registry
  - Global access for all services

- **Persistence Service**
  - Configuration Admin is too complicated
  - Simple embedded database or interface to it
  - Object relational mapping
  - Support for simple transactions (single TX)
  - Data synchronization service

- **Communication Service**
  - Central administration point for communication channels
  - Simple selection of appropriate comm channel
  - Support of ad hoc networks

- **Webservice support**
  - Data transfer via web services (SOAP, WSDL)
  - Optimization for low bandwidth and unstable communication links
  - Use Model Driven Development for generation of data object and client stubs

- **Security**
  - Secure communication and data transfer
  - Link software to a defined device
  - Signature of software (bundles)
  - Confidentiality
  - Integrity protection
  - Structure security manager
  - Secure logs
  - Secure persistency
  - DRM an issue?
  - Back data, configurations etc. to a device
  - Flexible HMI
  - Updatable during application life time
  - Provide an API
  - How to interact with the user?
  - Customizable

- **HMI**
  - Flexible HMI
  - Updatable during application life time
  - Provide an API
  - How to interact with the user?

**Non Functional**

- **OSGi what?**
  - Sharpen the view
  - Define clear goals for VEG
  - Prioritize it
  - Which domains do we want to work in?

- **Ubiquitous**
  - Many supported HW/OS platforms
  - Becoming an industry standard
  - Harmonization with JCP/JSR, Eclipse etc.
  - Harmonization with current JVM version

- **Tools**
  - Better tool support could make OSGi more attractive for developers and management
  - Configuration of bundles, settings and other "attributes"
  - Generation of interfaces from UML (MDA/MDM)
  - Debugging
  - Cross Platform Development

**Automotive Industry**

- What about AMI-C use cases?