eclipseCON™ 2008

Eclipse-based Embedded Engineering Environments Inspired by AUTOSAR

Stephan Eberle, Geensys

eclipseCON™ 2008

About…

- Stephan Eberle
  - Paris, France
  - Development lead of Geensys' AUTOSAR Builder product
  - Committer for EMFT Teneo component
  - 10+ years experience in OO and automotive embedded
  - Frequent presenter at conferences and events
- Geensys
  - **G**lobal **E**mbedded **E**lectronics & **N**etworked **SY**stem **S**olutions
  - Embedded system engineering and IP
  - Embedded system engineering tools
  - Consulting services

eclipseCON™ 2008

Outline

- **AUTOSAR from 4000 meters**
- Need for custom AUTOSAR tools
- Definition of an open AUTOSAR tool platform
- Conclusion

eclipseCON™ 2008

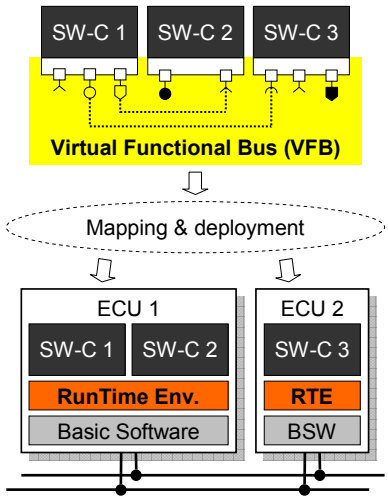
Objectives of AUTOSAR

= **AUT**omotive **O**pen **S**ystem **AR**chitecture

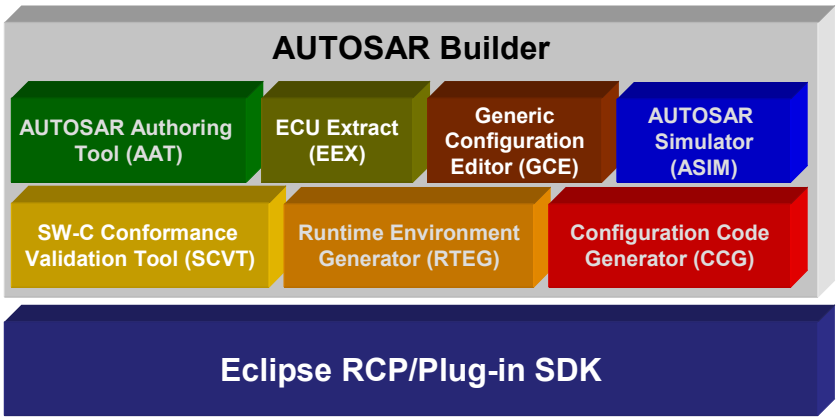
- Mastering the ever increasing number of functions and complexity of E/E systems
  - Getting out of “one function – one ECU” trap
  - Transparent collaboration between OEMs and suppliers and reduction of integration pain
  - Reuse of E/E functions
    - In different product lines
    - With different hardware platforms/communication systems

How it works

- Decoupling of application software from ECU hardware and communication systems through VFB/RTE
- Formal (XML-based) description of
  - Application software components
  - ECU hardware
  - System topology
- Arbitrary mapping of application software components to ECUs
- Formal (XML-based) configuration of standardized basic software modules
  - Operating system
  - Device drivers
  - Communication drivers



Geensys' Eclipse-based AUTOSAR Tooling



eclipseCON™ 2008

To do list for AUTOSAR adaptors

- Coping with migration issues, e.g.
  - Import/conversion of legacy description formats
  - Conversion between AUTOSAR releases
- Handling of domain/vendor-specific aspects within AUTOSAR, e.g.
  - Automatic configuration of BSW modules from ECU Extract
  - Restriction of AUTOSAR design activities wrt custom development process and roles
- Integration with non-AUTOSAR development tools, e.g.
  - Model-based design tools
  - Build tool chains

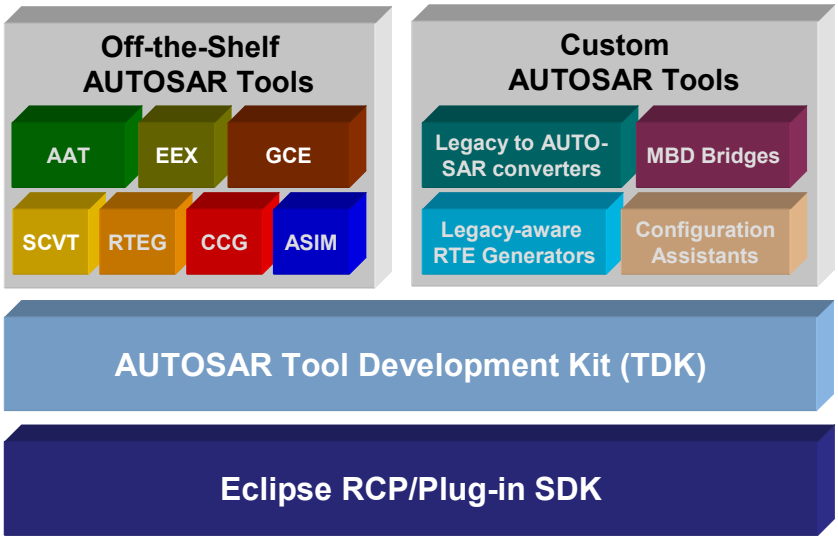
eclipseCON™ 2008

What does all that mean to AUTOSAR tooling?

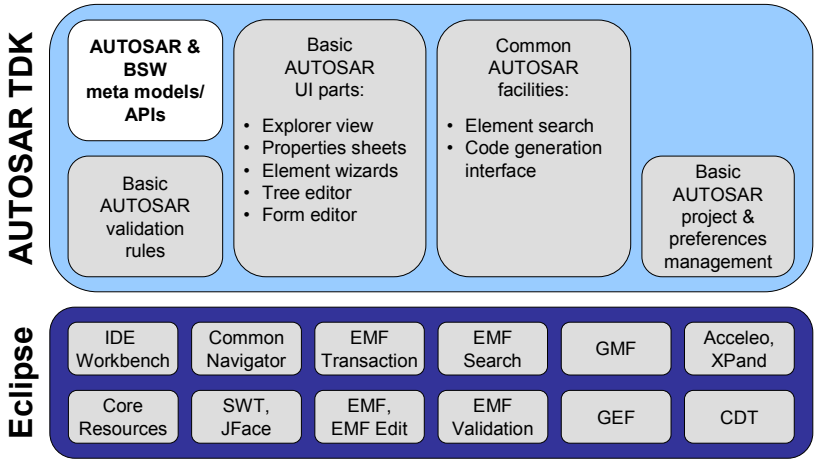
- AUTOSAR tool environments
  - Will never be complete out of the box
  - Must be highly adaptabe to domain/vendor-specific contexts
- Idea
  - Provide an **open and extensible AUTOSAR tool platform** as common core for off-the-shelf and custom tool components
  - Final environment must enable both **creating and using AUTOSAR tool components**

↳Solution:

Extend Eclipse Platform towards an AUTOSAR tool platform



AUTOSAR TDK Components



Conclusion

- AUTOSAR is used in many different domain/vendor-specific contexts
- It is unlikely that a single AUTOSAR tool can satisfy all needs
- Providing an open and extensible tool platform is crucial for achieving the required degrees of **customizability**

