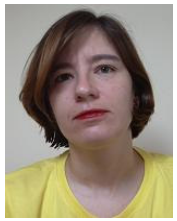
eclipseCON™ 2008

Integrating Mobile Devices and Emulators onto the Eclipse IDE with TmL

Christian Kurzke
TmL Team Lead



Mauren Brenner

Fábio Fantato

Daniel Franco

 | 18 March 2008 | Long Talk

Tools for mobile Linux (TmL)

- History
  - December 2006: creation review
  - March 2007: short talk at EclipseCon 2007
  - Source code available in CVS repository
- Goal
  - Provide support for development of mobile applications on the Eclipse IDE
- Related projects
  - CDT, all other DSDP subprojects (MTJ, NAB etc.)

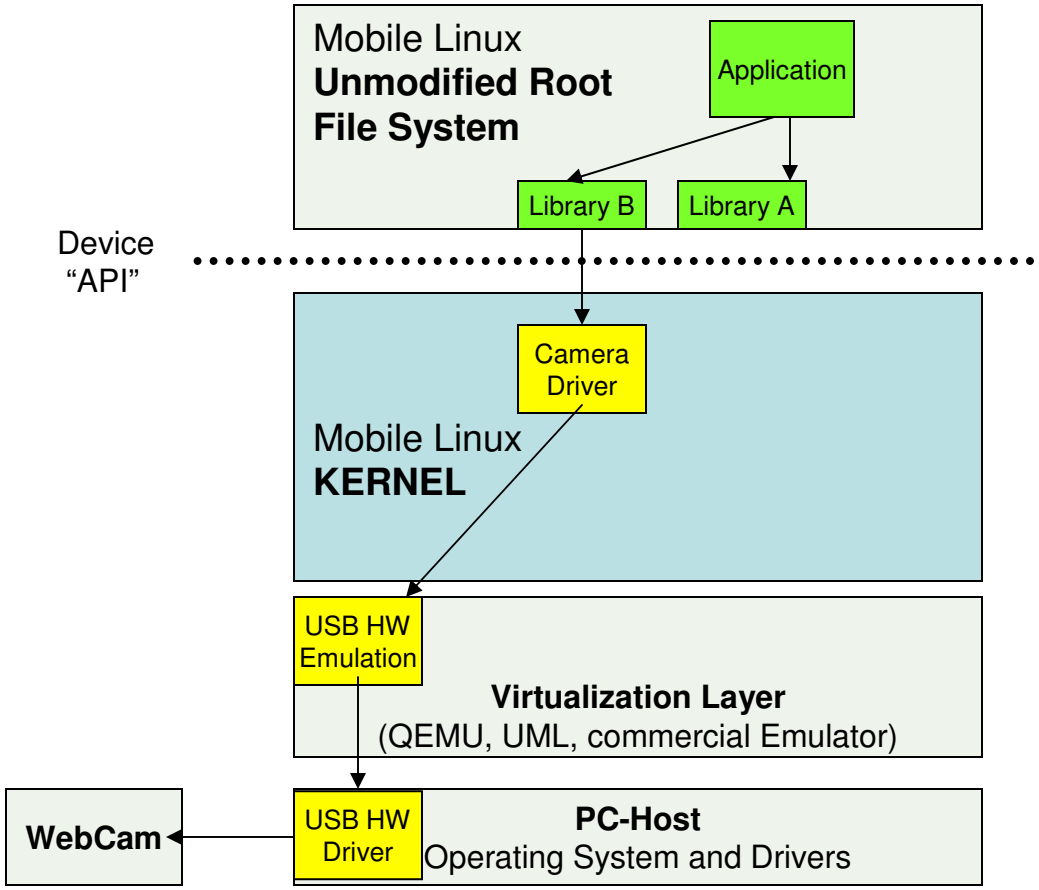
TmL - The Grand Vision

- Simulation of Specialized Hardware
  - "PC Type" hardware like
    - Bluetooth
    - Camera
  - "Mobile Device" specific hardware like
    - GPS
    - GSM/3G radio
    - Accelerometer
    - Orientation Sensors
    - Telematics
- Simulation of Infrastructure
  - Messaging services
  - Network services
  - Broadcast services

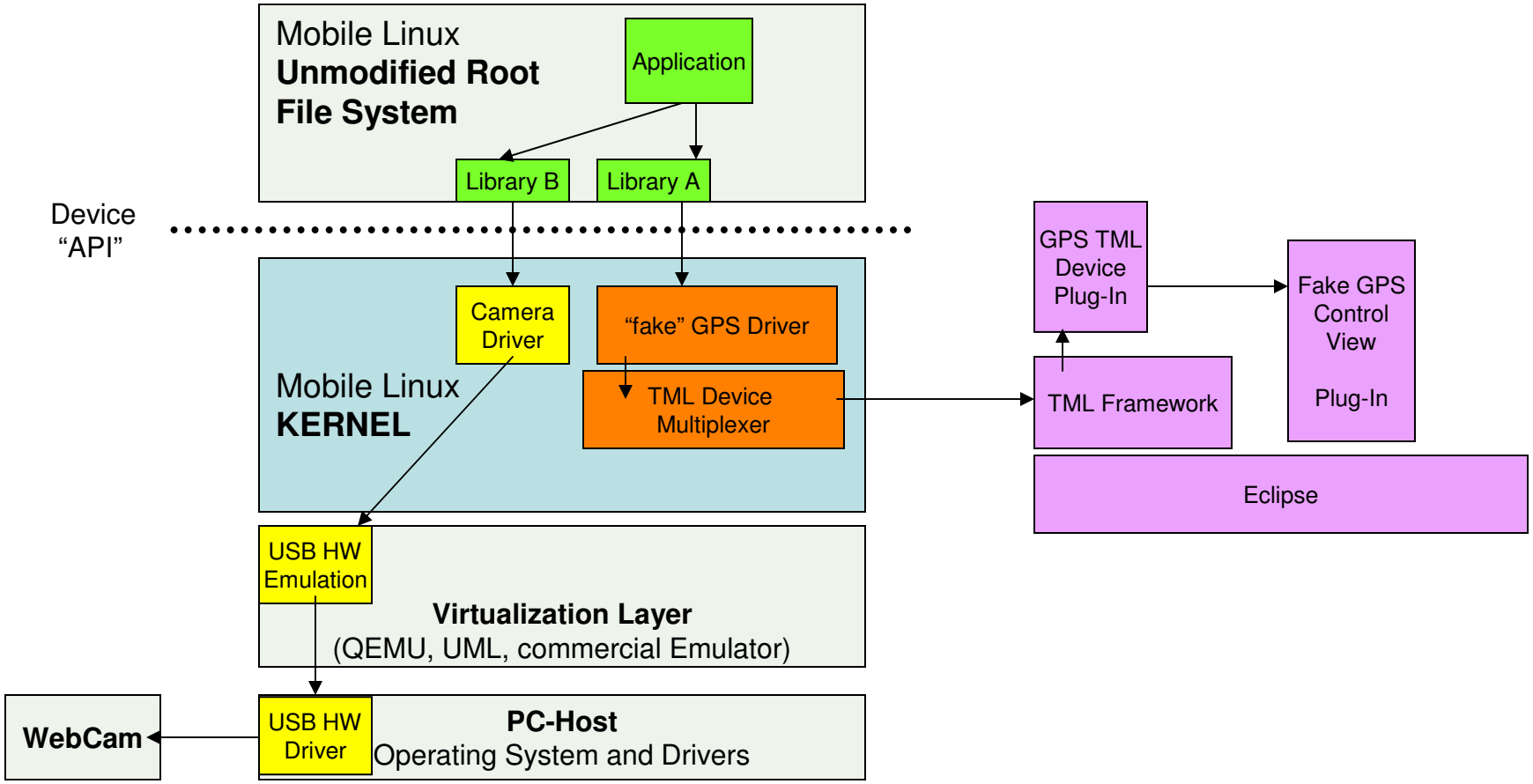
Building Blocks of TML

- All those devices can be modeled as “State Machines”
- TmL has a framework to define those state-machines
  - Define states and transitions in XML
  - Register code to execute for state transitions
  - Register code for control flow
- Simple State Machine for TML implemented and used for “VNC Viewer”

Simulated Device Framework (1)

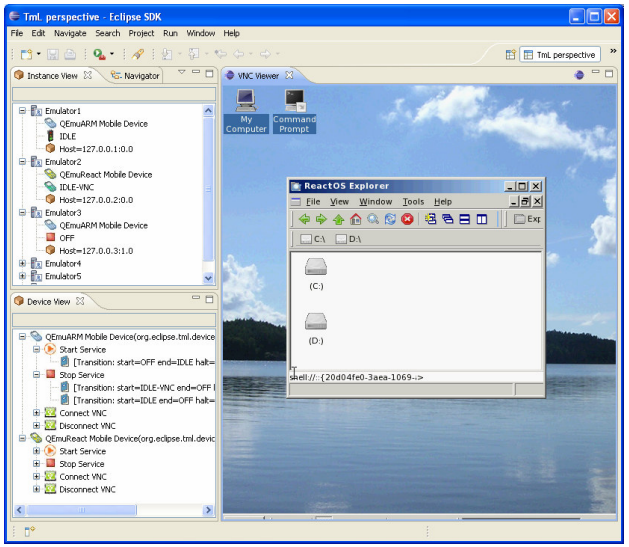


Simulated Device Framework (2)



Project Scope

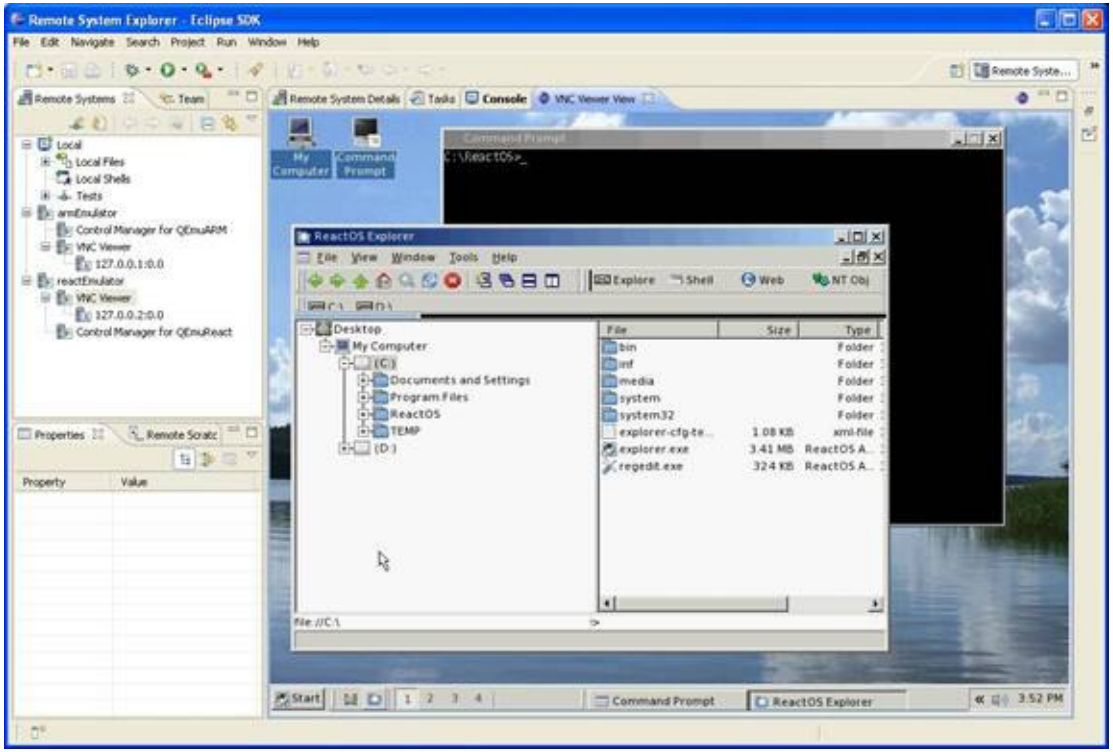
- Current scope
  - Device Framework supporting devices and emulators
  - VNC Viewer for display visualization



- Future scope
  - Simulated end-to-end environment

TmL and TM *(future)*

- TM: Target Management, another DSDP subproject
  - RSE: Remote Systems Explorer
  - Adapters

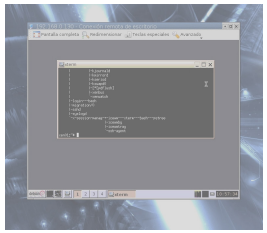


TmL Components

- Device Framework

  - Integration of devices and emulators to Eclipse IDE
  - Supports real, physical devices
  - Supports device emulators

- VNC Viewer



  - Graphic display visualization capabilities
  - SWT component
  - VNC client (VNC protocol, also known as RFB)

Device Framework

http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.device/?root=DSDP_Project

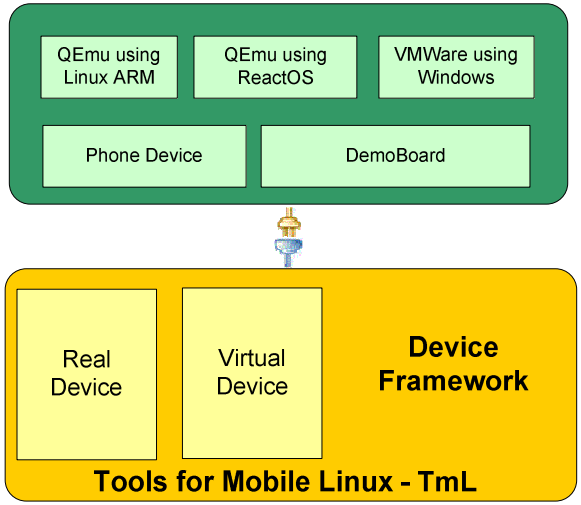
- Generic framework
  - Extension point mechanism
  - Generic classes and interfaces
- Target users
  - Device vendors
  - Emulator developers
  - SDK developers, where SDKs often include emulators
- Sample implementations

Definition of “Device”

- A device means a piece of hardware with a certain purpose or functionality, e.g. a mobile phone, a set-top box, a single board computer etc

- Abstract description of a device or emulator
  - Real, physical device
  - Emulator

- Device plug-in
  - Device or emulator properties
  - Provides a wizard to create instances
  - Contains components used by all instances
    - Scripts
    - Binaries

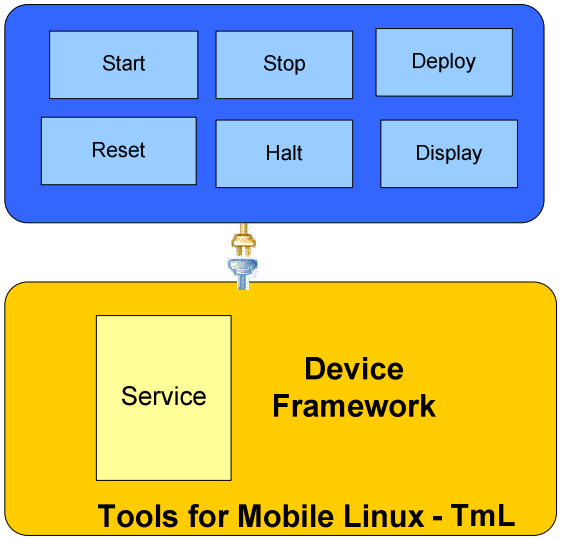


Definition of "Service"

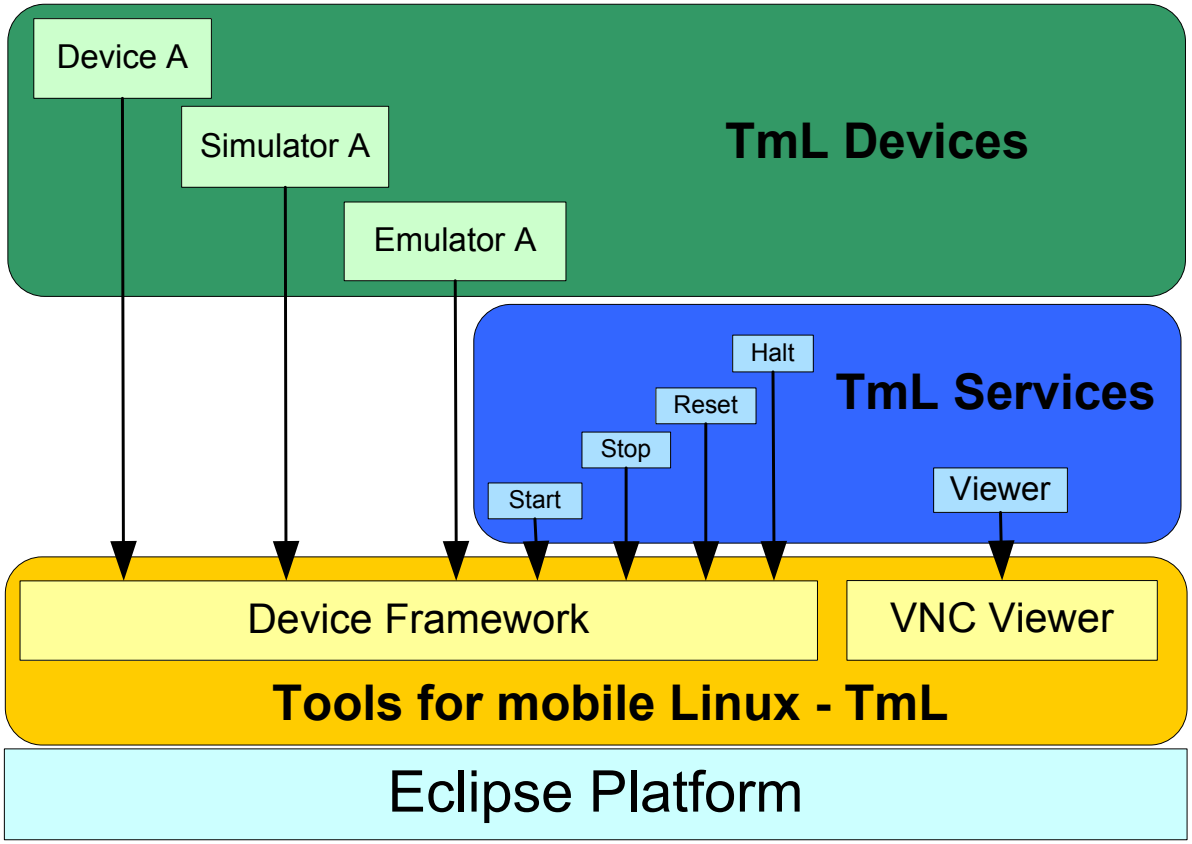
- A service is the implementation of functionalities or capabilities offered by each device

- Framework can provide common services
- Device services: start, stop, reset, halt, flash
- Emulator services: start, stop, restart, deploy

- Service plug-in
  - Contains service-specific components
    - Scripts
    - Wizards



Device Framework Architecture



Extension Points

org.eclipse.tml.device

org.eclipse.tml.service

org.eclipse.tml.serviceDefinition

org.eclipse.tml.state

Device Extension Point

org.eclipse.tml.device

Extension Element Details

Set the properties of "device". Required fields are denoted by "*".

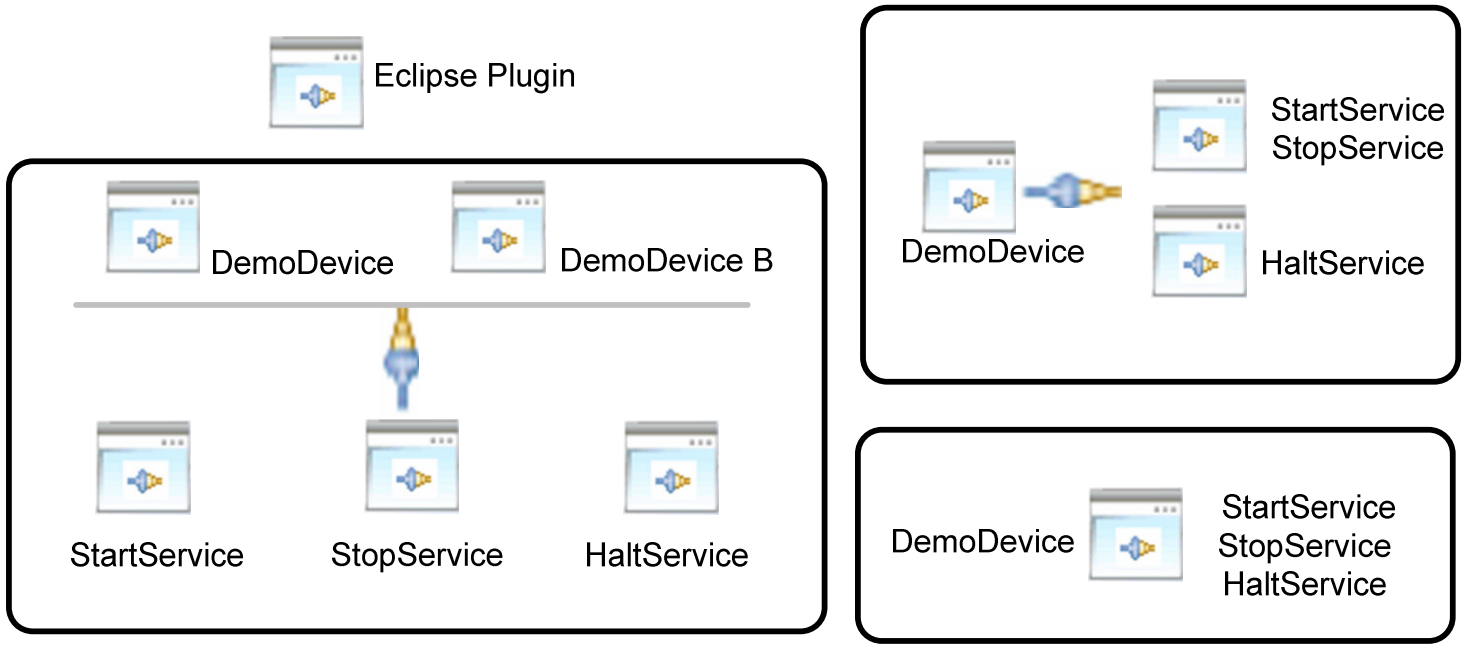
id*:	org.eclipse.tml.device.qemureact.qemureactDevice
name*:	QEmuReact Mobile Device
description:	Mobile Emulator for QEMUARM
version:	0.2.0
provider:	Eclipse.org
copyright:	Motorola Inc. 2007
handler:	org.eclipse.tml.device.qemureact.handler.QEmuF Browse...
icon:	icons/full/obj16/qemureact.gif Browse...

Service Extension Point

org.eclipse.tml.service

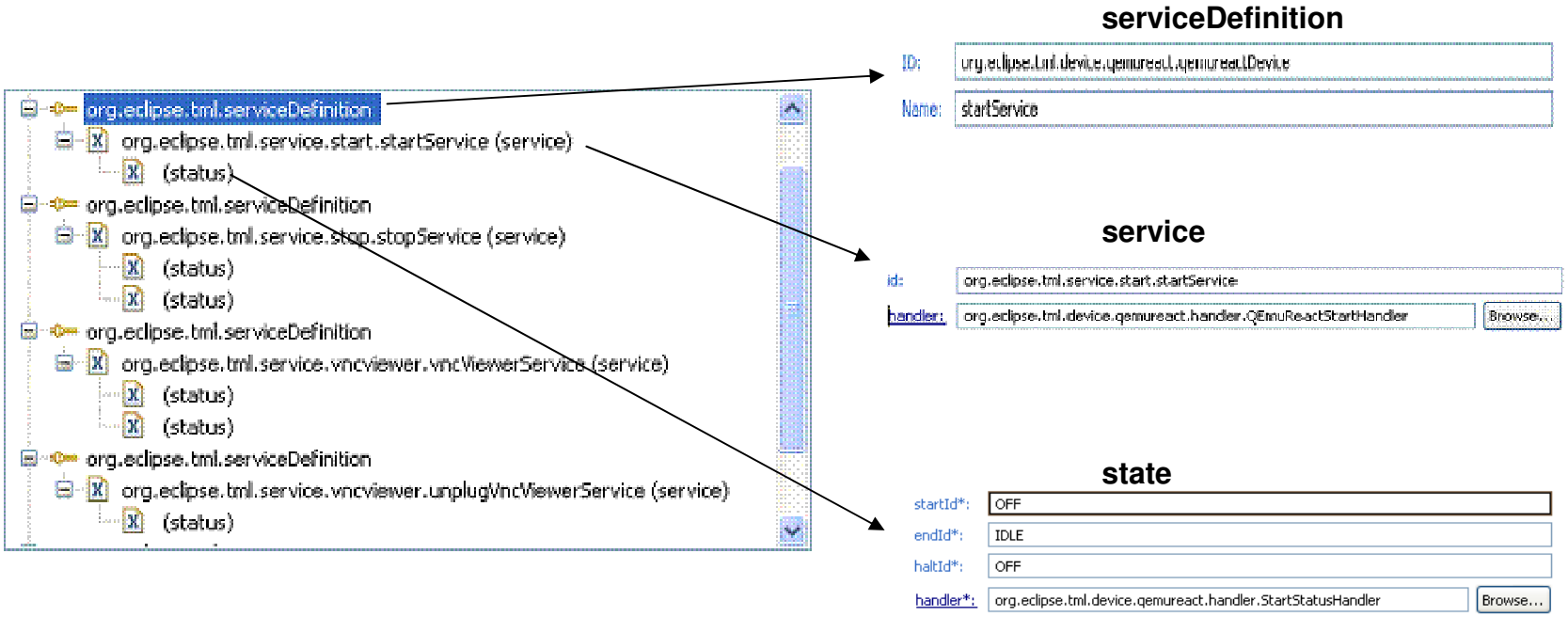
id*:	org.eclipse.tml.service.stop.stopService
name*:	Stop Service
description:	Service to Stop a mobile device
version:	0.2.0
provider:	Eclipse.org
copyright:	Motorola Inc. 2007
handler:	org.eclipse.tml.service.stop.handler.StopService [Browse...]
icon:	icons/full/obj16/stop.gif [Browse...]

Device and Service Plug-ins



Service Definition Extension Point

org.eclipse.tml.serviceDefinition

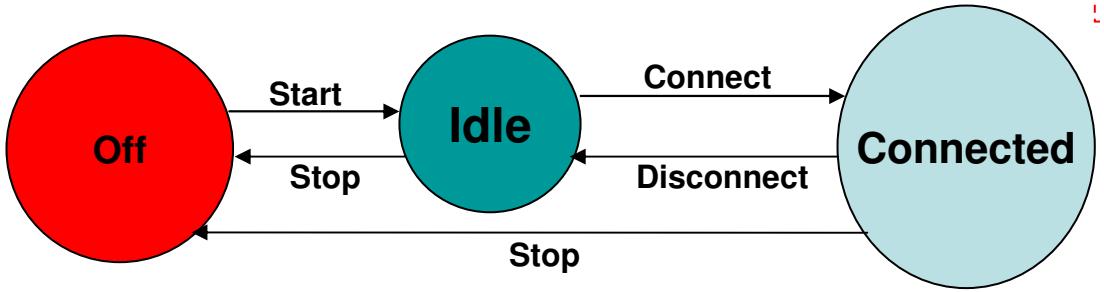


State Extension Point

org.eclipse.tml.state

example services

DemoDevice
StartService
StopService
ConnectService
DisconnectService

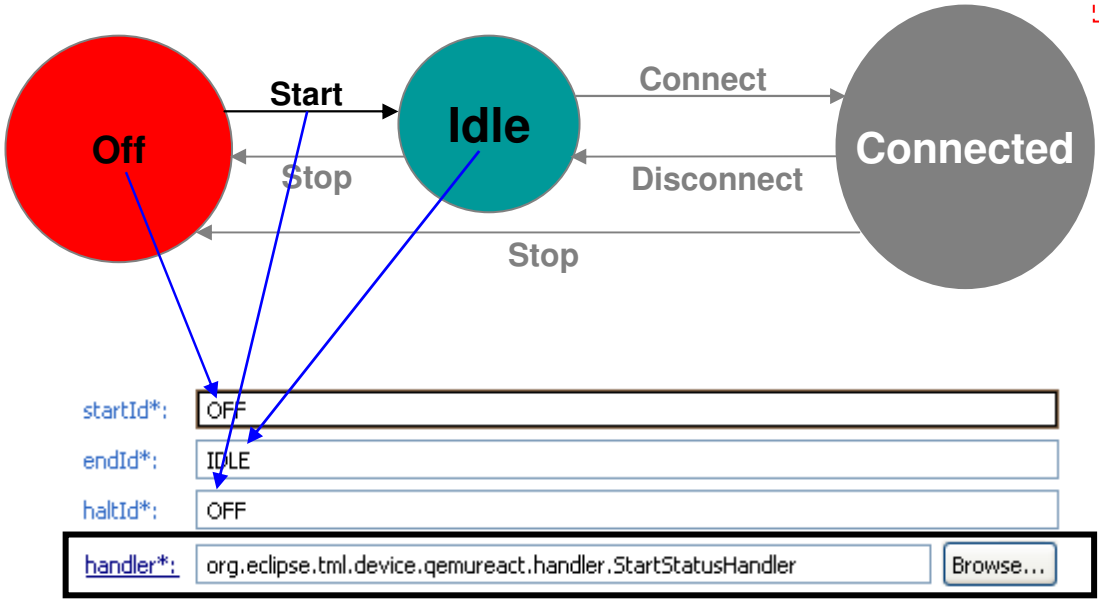


- Example states
  - *Off*
  - *Idle*
  - *Connected*
- State extensions
  - Each developer can create one's own set of states

State Transitions

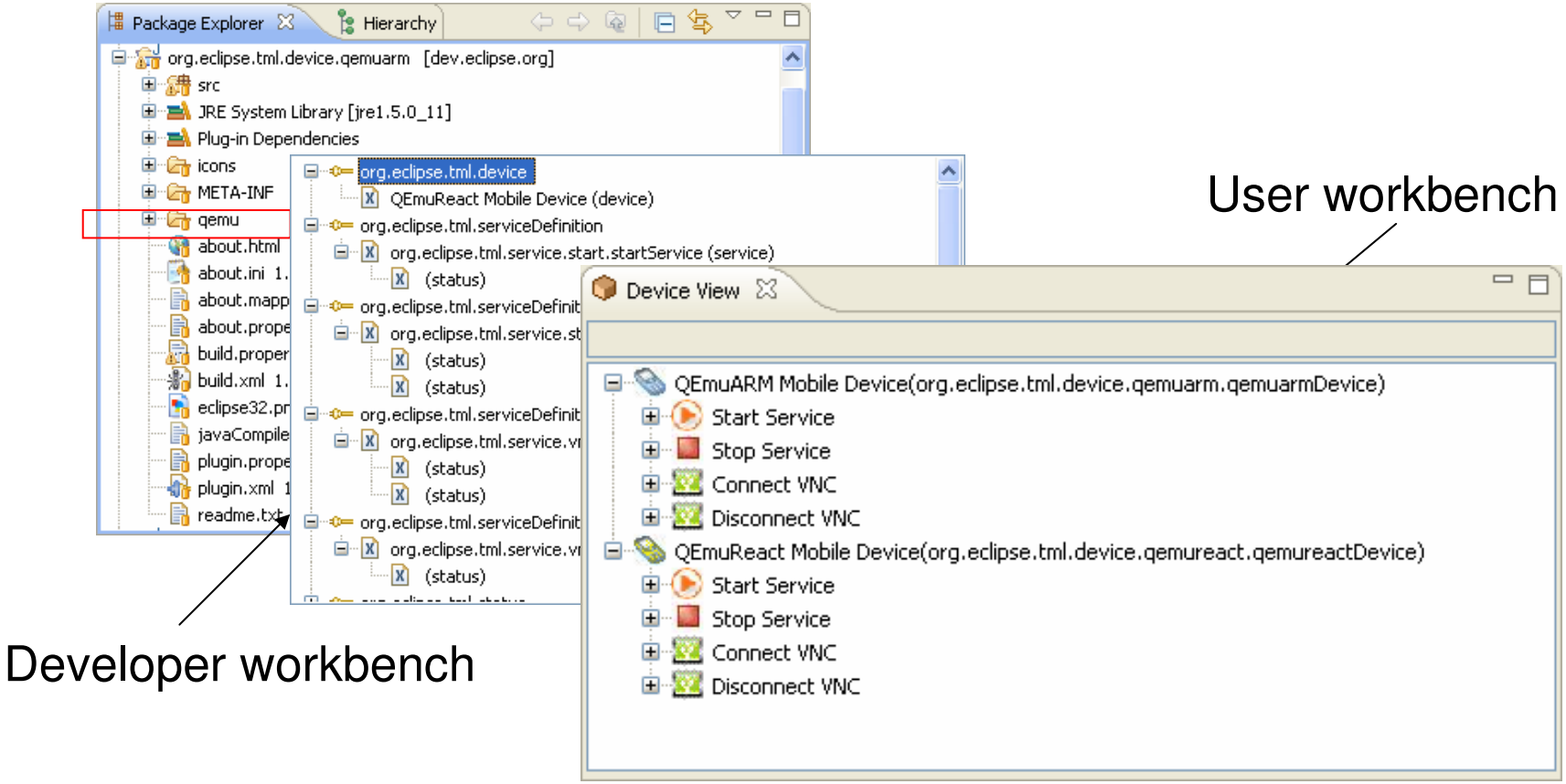
org.eclipse.tml.serviceDefinition

DemoDevice
StartService
StopService
ConnectService
DisconnectService

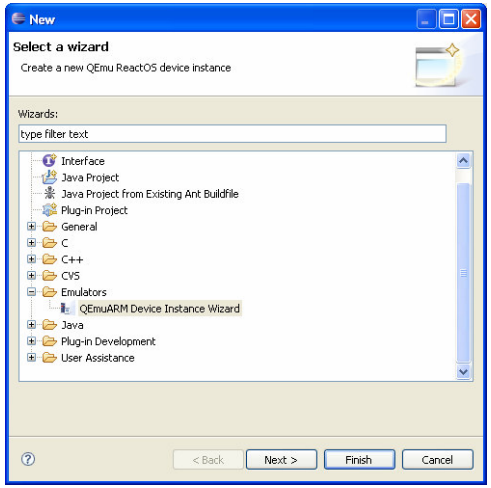


- There is a set of state transitions for each device and service

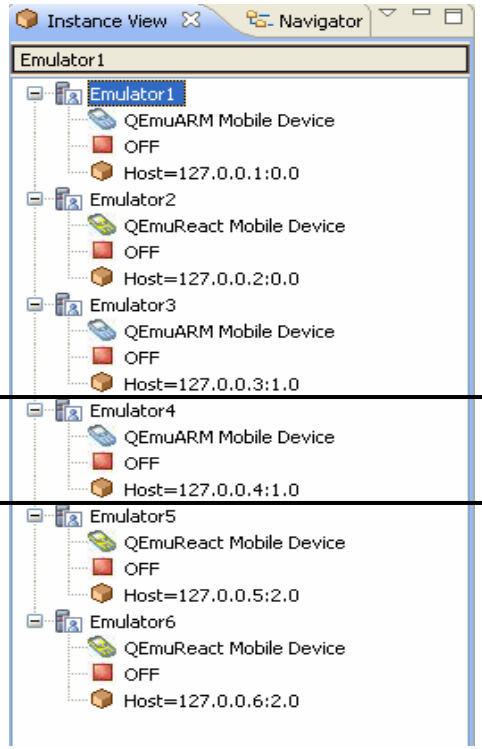
An Example Device Plug-in



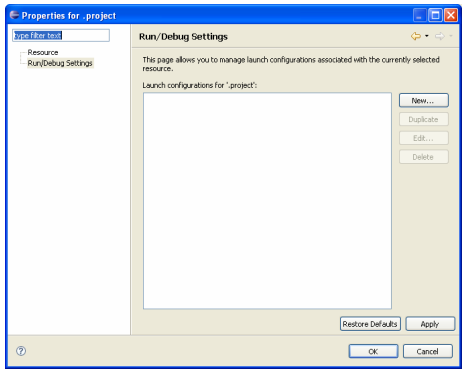
Creating Instances of a Device Plug-in



Instance creation wizards

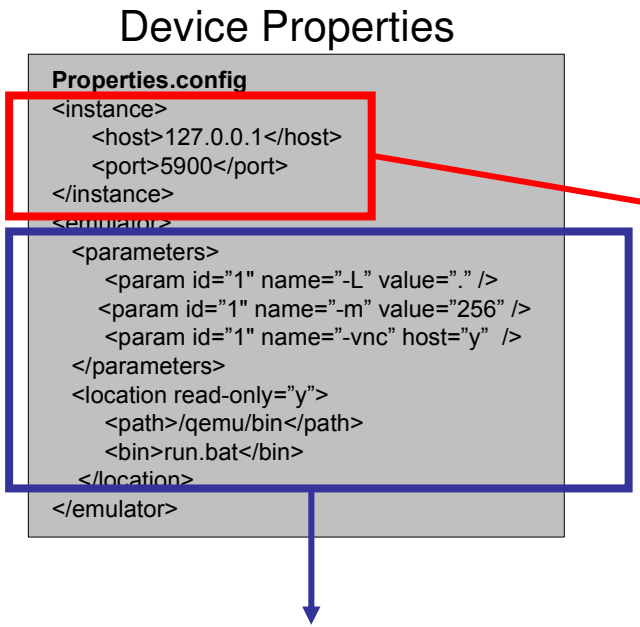


Instance view



Instance properties page

Device Properties and Instance Properties



Instance properties

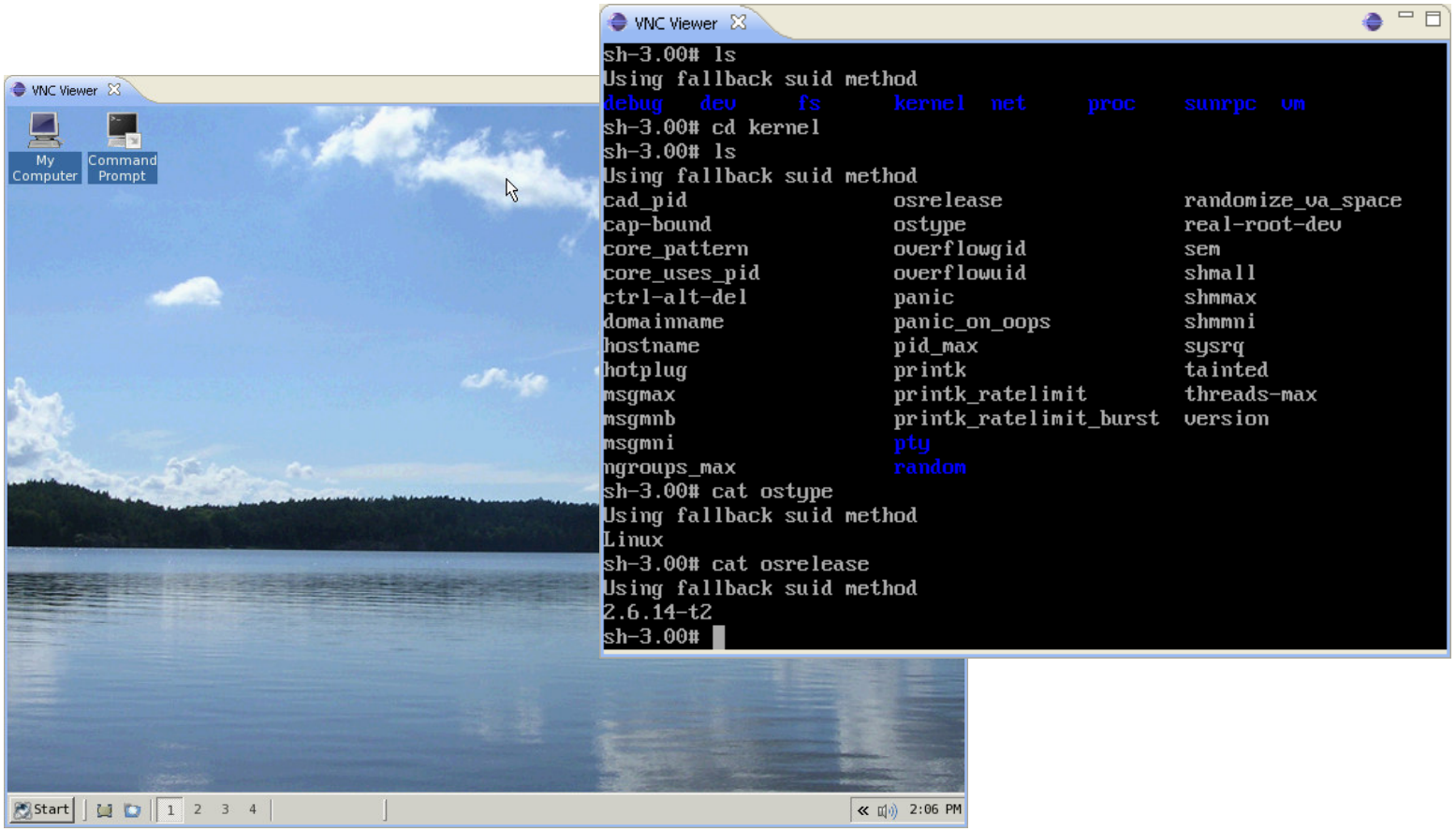
Properties.config
<host>127.0.0.1</host>
<port>5900</port>

- Rules according to XML definition
- Editable using Properties page
- Saved as metadata in workspace

- Editable using Preferences page
- A set of properties per device plug-in
- Rules according to XML definition

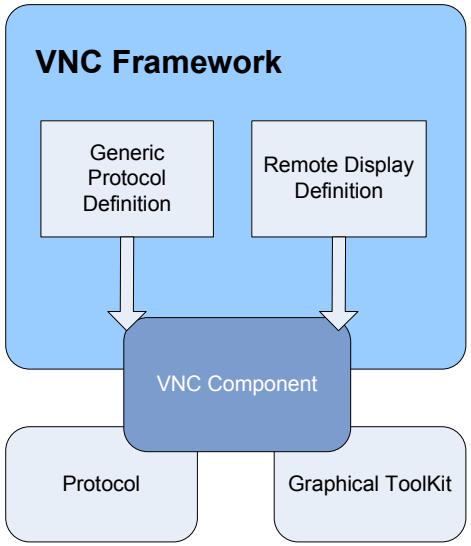
VNC Viewer

http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.vnc/?root=DSDP_Project

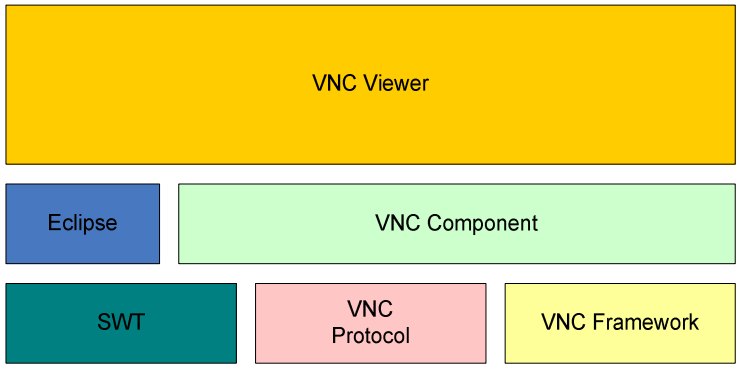


VNC Viewer Architecture

- SWT component
  - Graphical display viewer
  - Can be used standalone or within an Eclipse view
- VNC client
  - VNC protocol (or RFB, Remote Frame Buffer)
  - “Feeds” the SWT component



VNC Viewer View - Eclipse



VNC Viewer: Ideas for the Future

- Configurable skins
- Multiple displays
- Keyboard maps
- Extensible protocol

Simulated End-to-End Environment *(future)*

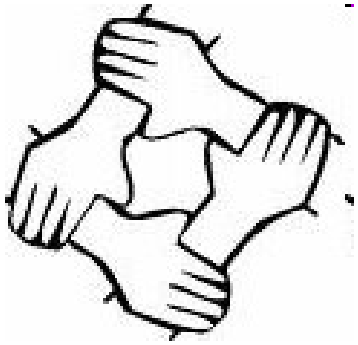
- Complete network infrastructure
- Connection among mobile devices as well as back-end servers
- Network nodes are devices and emulators implemented by means of the Device Framework
- Suitable environment to test mobile enterprise applications
- A potential testbed for innovative applications

Demo

http://wiki.eclipse.org/DSDP/TML/How_to_configure_TmL_demo

TmL Project Resources

Project web site:
http://www.eclipse.org/dsdp/tml
Project wiki:
http://wiki.eclipse.org/DSDP/TML
Users newsgroup:
eclipse.dsdp.tml
Developer mailing list:
dsdp-tml-dev@eclipse.org
CVS repositories:
http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.device/?root=DSDP_Project
http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.vnc/?root=DSDP_Project
TmL demo:
http://wiki.eclipse.org/DSDP/TML/How_to_configure_TmL_demo



Suggestions and contributions are welcome! ☺

Questions & Answers