The DSDP Target Management Project

Martin Oberhuber, Wind River
www.eclipse.org/dsdp/tm

Agenda

- TM on Europa
  - Online Demo
  - TM 2.0 New Features
- TM for Extenders
- TM Future Planning

The Eclipse Target Management Project

… why “Target”?

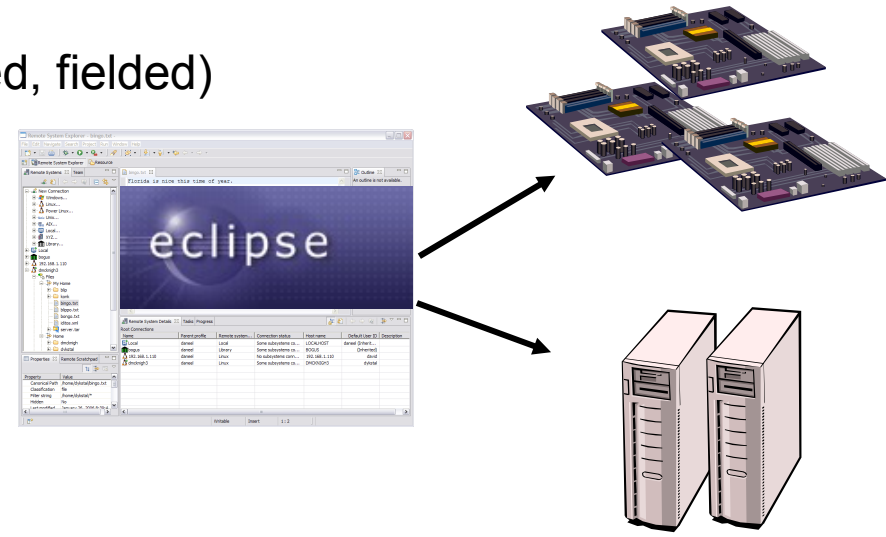
- Just a matter of terminology

Remote Computer Systems

- Targets (Locally connected, shared, fielded)
- Hosts (Grids, farms, nodes)

and running software on them

- Discover, connect, get status
- Download, run, debug, test

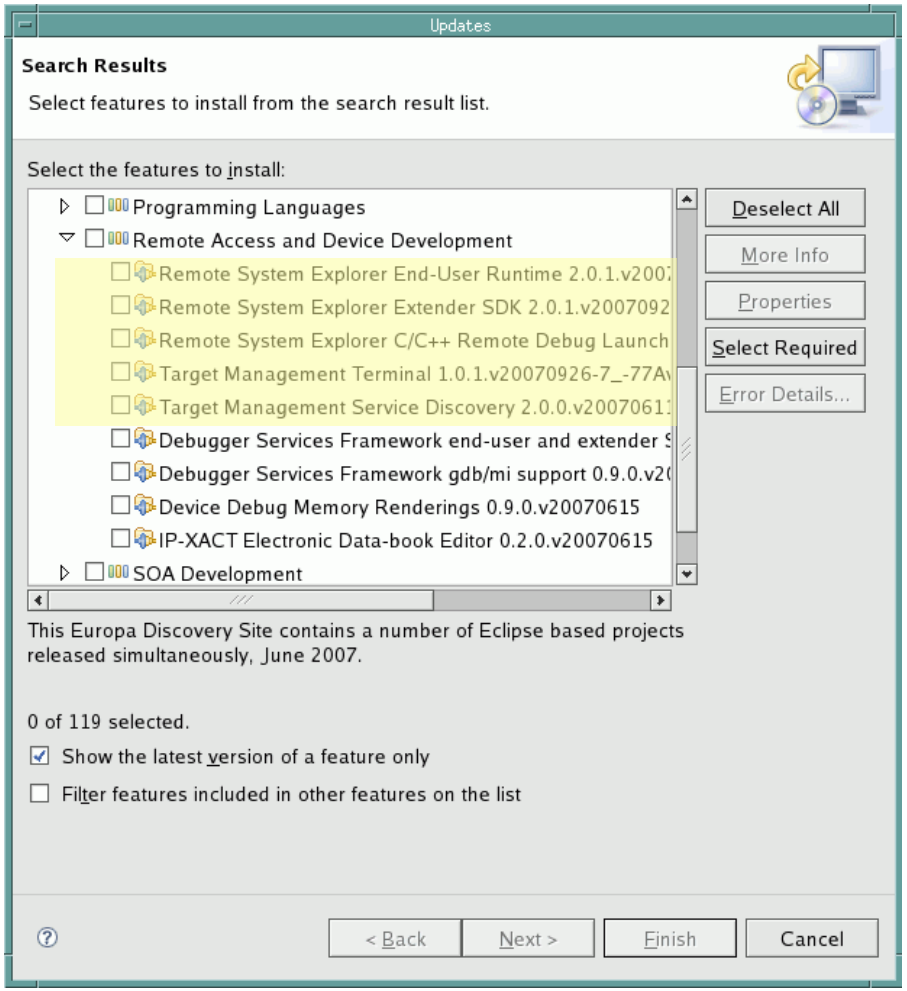


… why “Management”?

- Discover remote systems; manage their properties and capabilities; team-share connection definitions and settings; access control

Target Management 2.0 on Europa

- End-User Tools
  - Remote System Explorer (RSE): Transparent remote file access
  - ANSI Terminal
  - CDT remote debug
- Components for Extenders
  - Service Discovery



TM as an out-of-the-box tool

"I installed the RSE and it works exactly as expected. Using the SFTP connection method, I can connect to my servers, browse my files, open and edit PHP, HTML and other various text files using my favorite editor tools seamlessly in Eclipse as though they were local. Opening the file triggers a download, saving the file triggers an upload."

Denis Roy, Eclipse Webmaster,
http://eclipsewebmaster.blogspot.com/2007/01/remote-editing-using-eclipse.html

Remote System Explorer - test2.c - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Remote Systems | Team

- Local
- build.eclipse.org
  - Sftp Files
    - My Home
    - Root
    - RSETest
      - dir2
      - linkReadme5.txt
      - readme5.txt
      - RSE-SDK-latest.zip
      - test.o
      - test2.c
  - Ssh Shells
- szg-rhel4
  - Files
  - Processes
    - All Processes

readme5.txt | test2.c

```
#include <stdio.h>
int main(int argc, char** argv) {

    printf("The RSE supports editing remote files\n");
    printf("just as if they were local.\n");

    printf("Compile commands can be executed remotely,\n")
    printf("and build errors are parsed to support direct\n");
    printf("navigation into the code.\n");

    printf("ISVs can extend the RSE easily.\n");
    return 0;
}
```

Subsystems manage resources of a particular kind

Filters select resources dynamically

Properties | Remote Scrat...

Property	Value
Executable Nan	tcsh
Group ID	100
Name	tcsh
Number of child	0
Parent PID	5738
Process ID	5741
State	S-Sleeping (Interruptible

Tasks | Remote Shell

build.eclipse.org

Command Shell - Running

```
/home/data/users/moberhuber/rsetest>
gcc -o test2 test2.c
test2.c: In function `main':
test2.c:8: error: parse error before "printf"
/home/data/users/moberhuber/rsetest>
```

Command

Remote System Details

Remote system filter My Processes

Name	Process ID	VM
su	1089	4848 kilo
gconfd-2	5138	5584 kilo
xterm	5738	10188 kilo
tcsh	5741	5472 kilo
java	7556	218976 kilo
xterm	13436	10856 kilo
tcsh	13439	7280 kilo

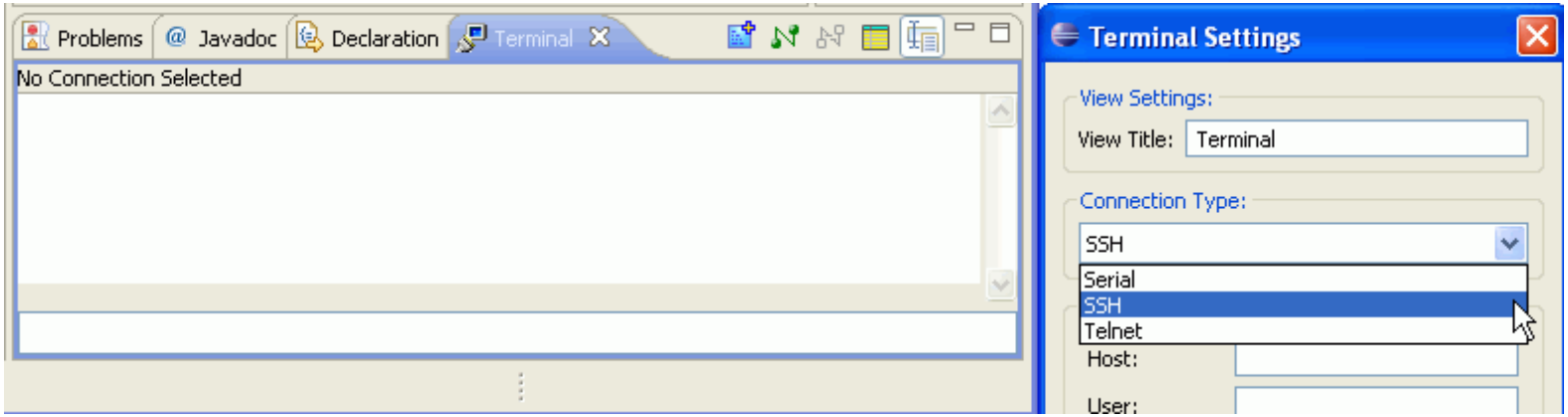
Process: tcsh

RSE Features

- Files Subsystem:
  - Remote Edit & Compare, with caching; Persistent, shareable filters
  - Tree and Table View, Drag&Drop
  - Search and Archive handling (dstore and local only)
- Remote Command (Shell) subsystem
  - Can be the basis for other subsysems
  - Line-based command/response (not a Terminal!)
  - Output filtering e.g. for compile commands
- Processes Subsystem
  - List and Kill processes on Linux and other UNIX

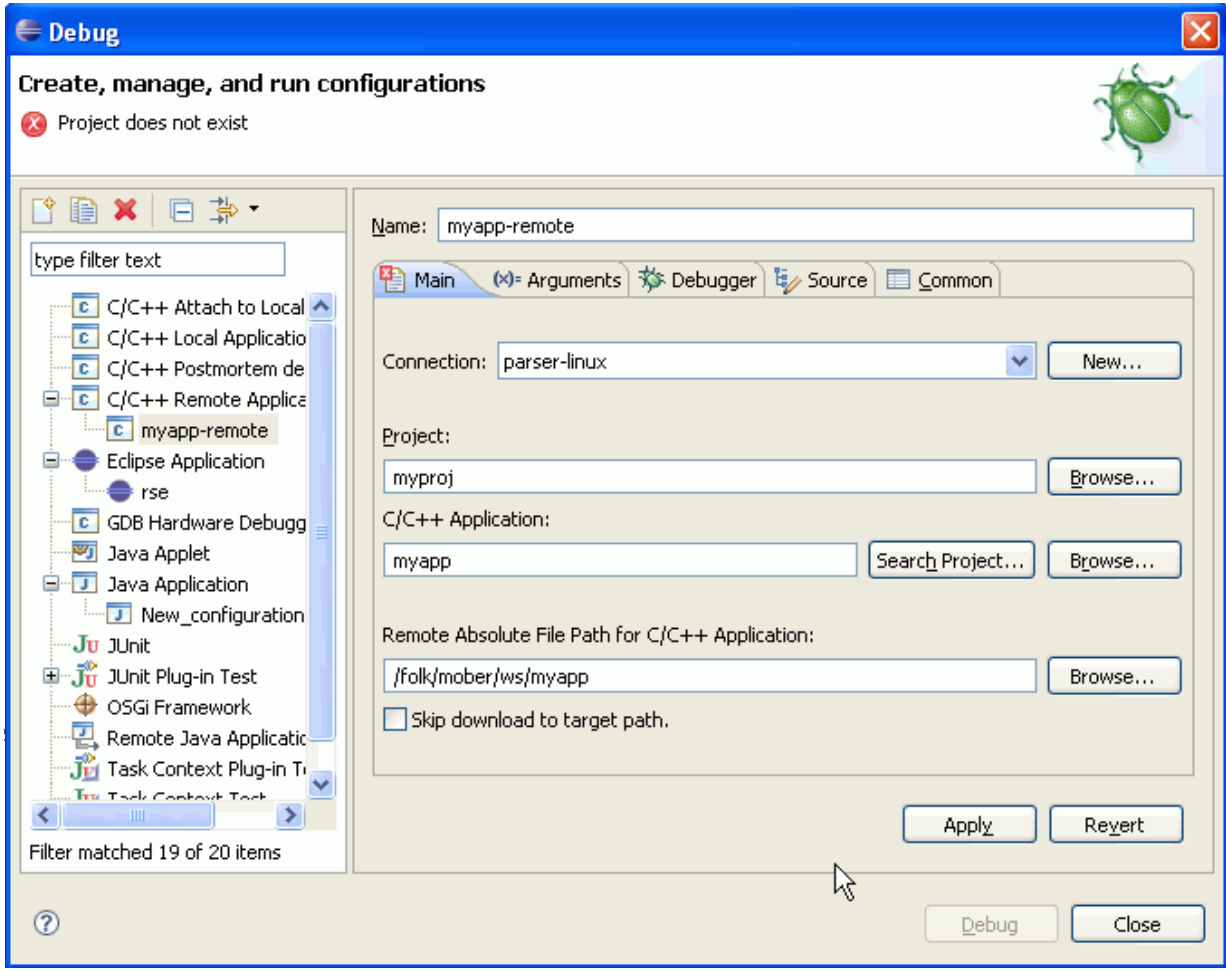
TM Terminal

- Window > Show View > Other > Terminal
  - (e)RCP embeddable widget for ANSI Terminal emulation
  - Cloneable View
  - Pluggable Connectors (Serial, SSH, Telnet)
  - New: Optional Input Line



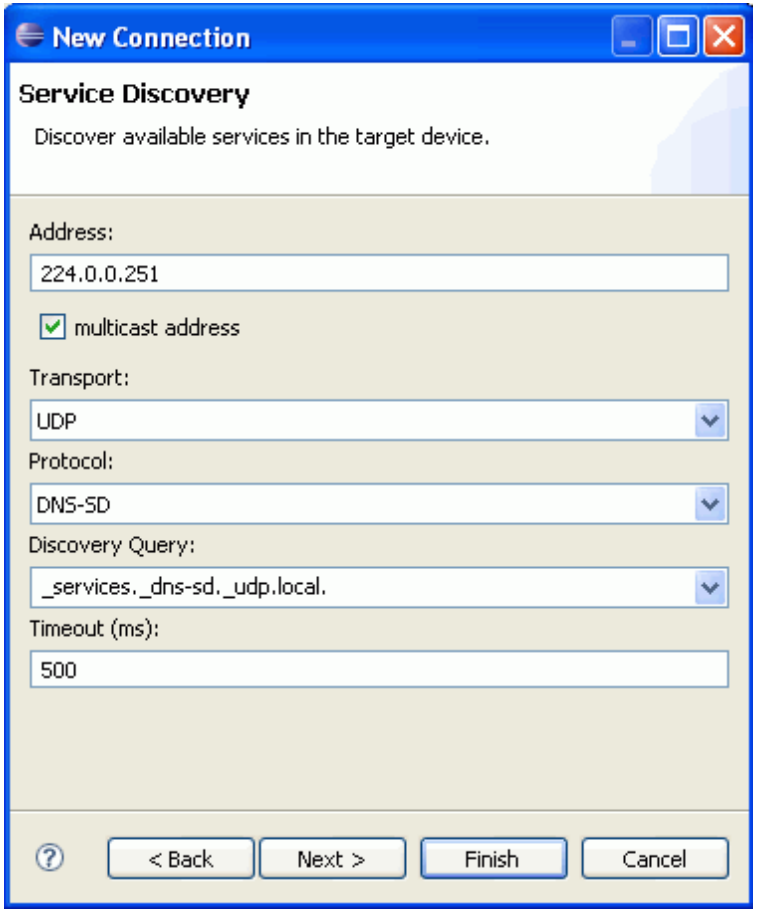
Remote CDT Integration

- Gdbserver protocol
  - RSE-contributed file upload
  - RSE-contributed shell channel
- There are also other possibilities for remote debugging, e.g. “ssh remote gdb” directly in CDT



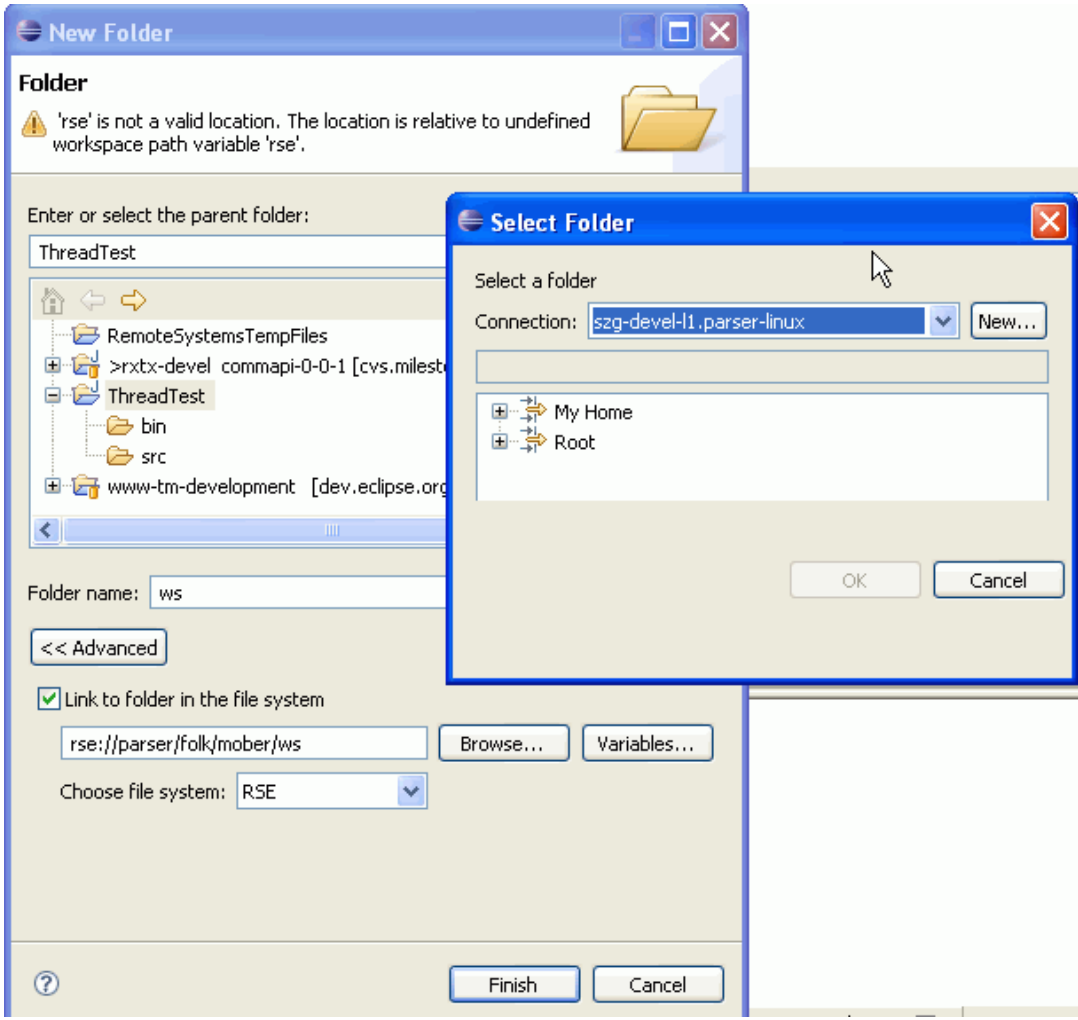
Service Discovery

- Find available ssh, telnet and other services through Zeroconf / DNS-SD
- Stand-alone or as an RSE Wizard: “New Connection: Type = Discovery”
- New: Auto-fill-in the multicast address



TM as an EFS Provider

- Eclipse New > Folder > Advanced > Link to File System > RSE
- Verified with ssh, ftp, dstore over JDT
- WST support for EFS recently added (Fall Maintenance) – for PHP



TM as an EFS Provider

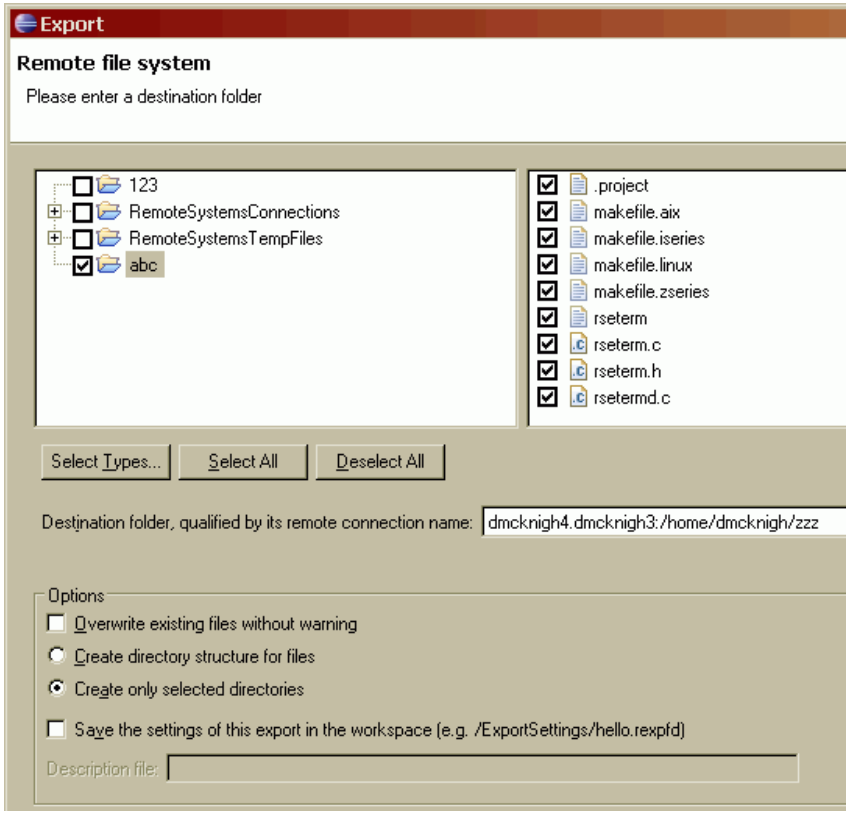
- Eclipse Filesystem (EFS) makes the Eclipse Resource System and Workspace very flexible, allowing any kind of (remote) resources to be shown in the Workspace
- TM as an EFS Provider supports any contributed file system
  - Out of the box: SSH, FTP, dstore – very simple API
  - Supports connection and password management
  - Supports directory contents caching where possible
- As workspace resources can now be remote…
  - … remote support is easily added to any tooling based on Eclipse, such as CDT, PHP, Web, …including parsers and outline views
  - … but the tooling needs a thin layer of awareness, i.e. access the Eclipse Resource model in a clean manner

http://tmober.blogspot.com/2007/04/target-management-m6-efs-and-webinar.html

RSE Import / Export Wizards

- File > Export > Other > Remote
- Save export settings (filters) in a *.rexpld script to replay
- Local development, synchronize to Remote

- Future: Use Platform Team/Synchronization framework



TM 2.0 New Features

- Externally Visible New Features
  - **Eclipse Filesystem (EFS) provider** with credentials management
  - Import/Export facility
  - Shell Processes subsystem (contributed)
  - Telnet subsystem (contributed)
  - FTP Listing Parser extension point
  - DNS-SD Discovery
  - Terminal Connector Framework (provisional), Optional Extra Input Line

- Internal API review and cleanup:
  - UI/Non-UI splitting, API/Non-API splitting
  - **More background jobs for remote access**
  - Improved flexibility for system type and action contributions
  - Persistence pluggable and outside workspace.
  - Improved File Service (Streams, setReadOnly, setLastModified)

TM for Extenders

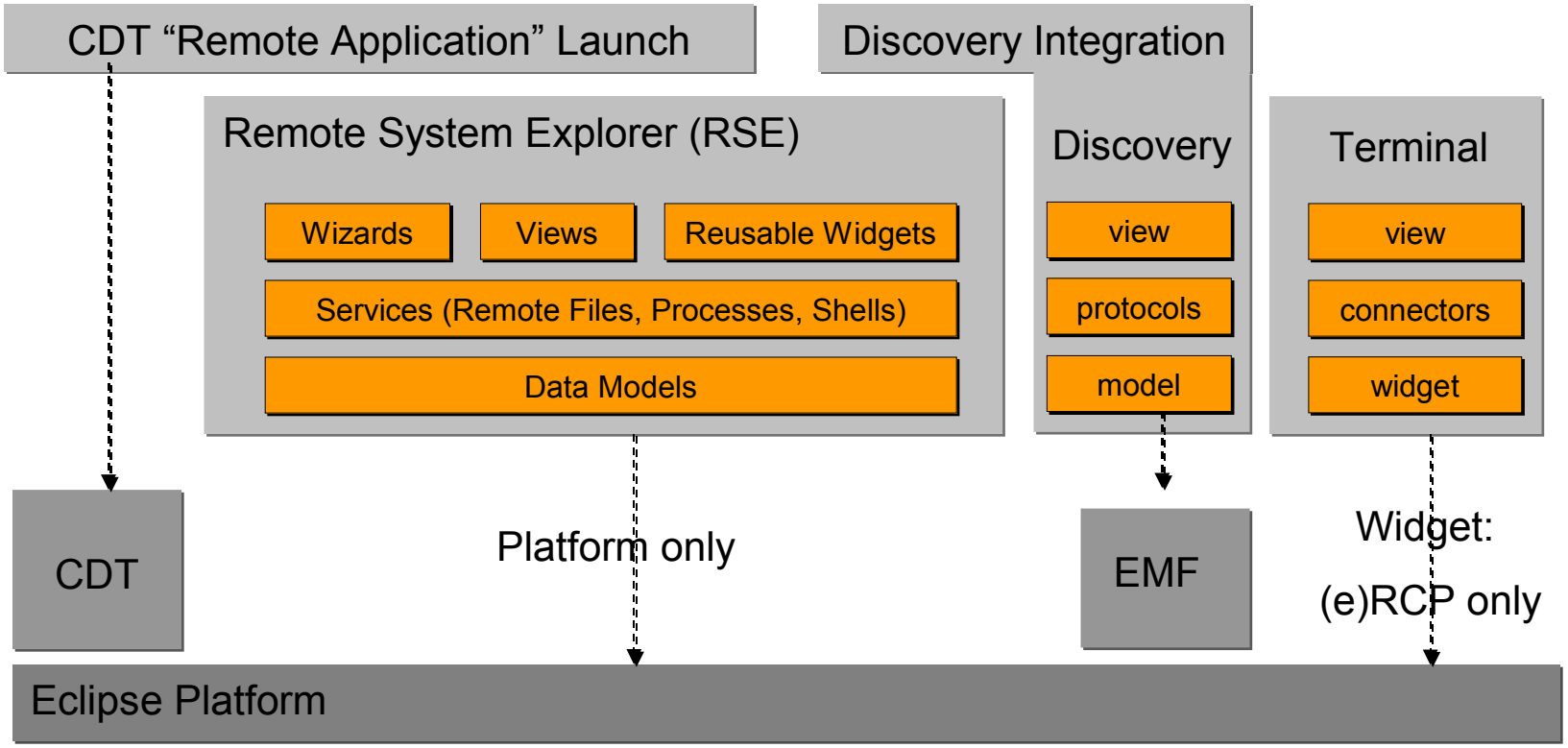
DSDP Target Management – Dates and Facts

- Major project milestones
  - Project Created – June 2, 2005
  - RSE 1.0 – Nov 12, 2006
  - RSE 2.0 – June 30, 2007 on Europa
- Continuing to expand community
  - EclipseCon tutorial – Mar 5, 2007
  - 2300 Downloads of RSE 1.0.1 – April 17, 2007
  - Increasing activity on Mailing List, Newsgroup, Bugzilla
  - Commercial adoption by at least 8 companies
- TM 2.0 project size
  - 290 kLOC (compared to TM/RSE 1.0: 242 kLOC)

TM Community

- Committers: Wind River (lead), IBM, Symbian
- Contributors: ACCESS, MontaVista, Tradescape, Celunite, Zylin
- Known Commercial Adoption:
  - Wind River Workbench 3.0
  - Atmel AVR32 Studio 1.0
  - EMAC Eclipse Distribution
  - Tradescape Clearing Tool 1.0
  - ACCESS Linux Platform Development Suite 2.0
  - MontaVista DevRocket 5.0
- Other active users: Ames DOE Lab, Broadcom, Cisco, Elastos, Festo Inc., LANL (PTP project), TmL Project (Motorola)

Target Management 2.0 Components



Target Management vs. RSE

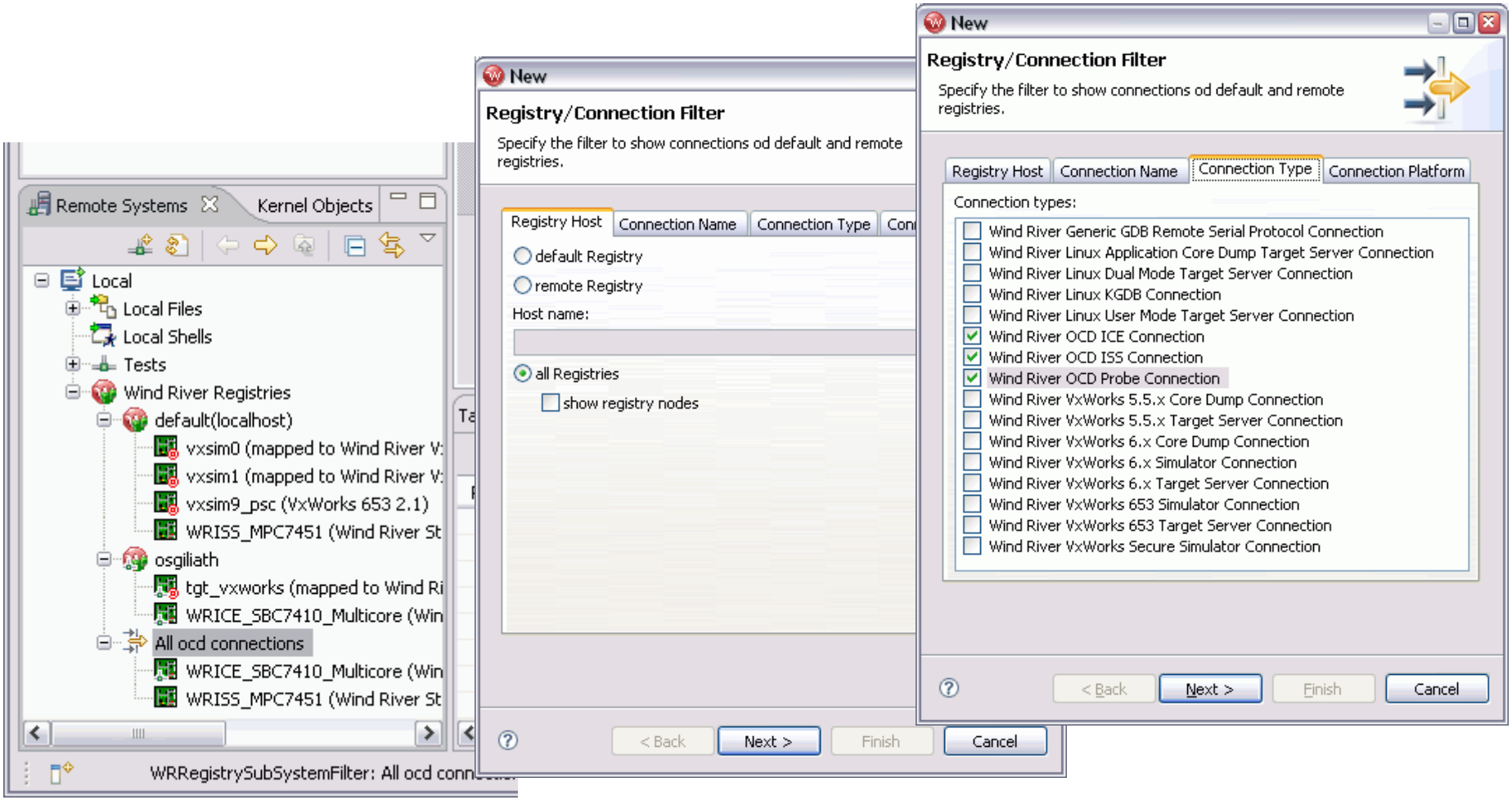
„The Target Management project creates data models and frameworks to configure and manage remote systems, their connections, and their services“.

- **org.eclipse.tm.core**: Core Components for remote access that can be re-used without other dependencies.
  - Jakarta Commons/Net 3rd party library
  - Discovery Framework and Zeroconf impl (needs EMF)
  - Wind River Terminal contribution
- **org.eclipse.tm.rse**: A consistent framework and UI for accessing remote compute resources from Eclipse.
- Remote System Explorer (RSE) integrates some (but currently not all) core components. **TM is the “project”, RSE is the “product”**.

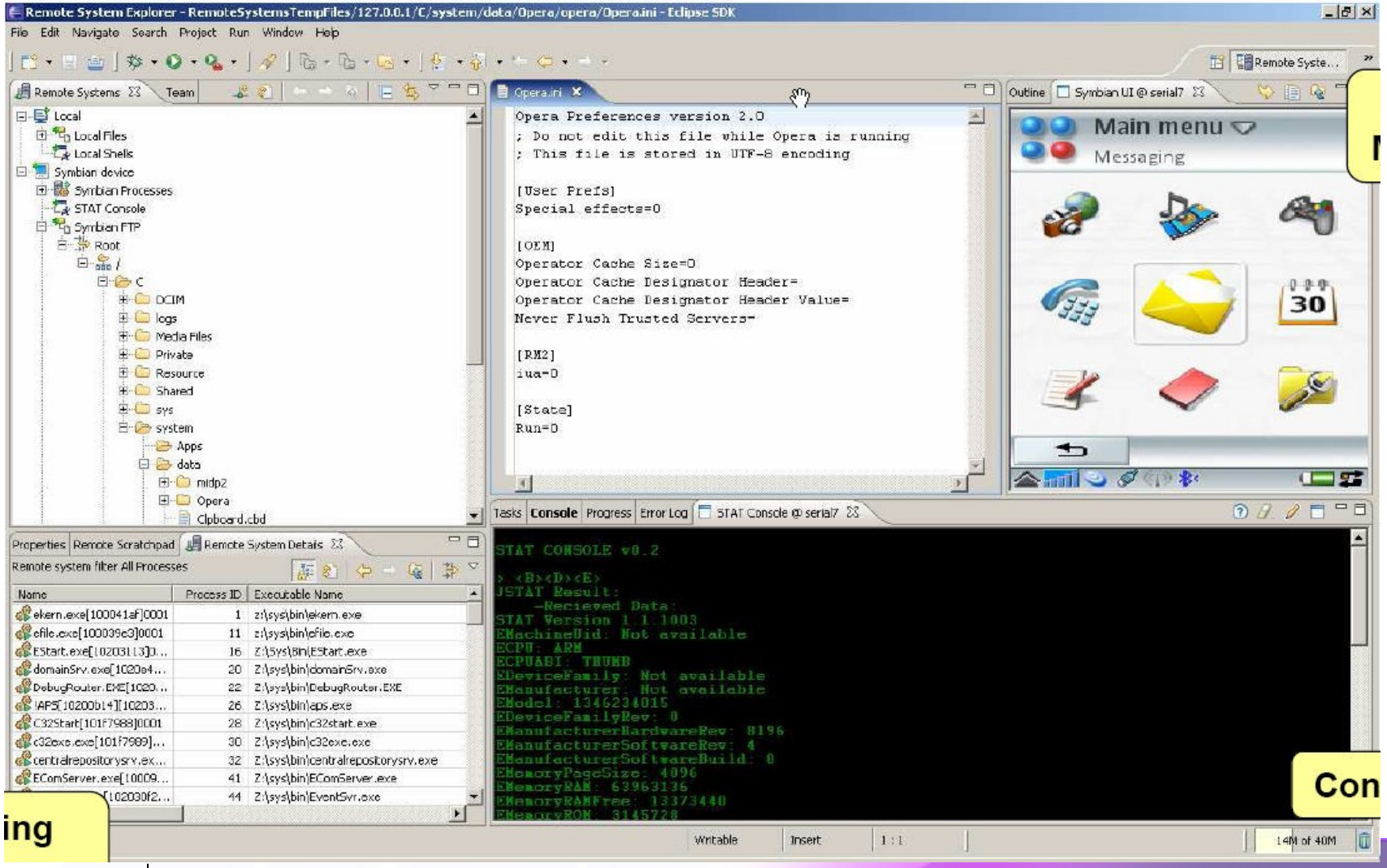
What do Extenders win by using TM?

- Common Core including persistence, filters, team sharing
  - Be open for 3rd party extensions through Open Source APIs
- Common UI including tree view, tables, monitor, browse dialogs
  - Access remote resources through a single, consistent UI
  - Few basic, re-used concepts (subsystems, filters)
- Common interfaces for abstract access to remote files, shells, processes through any contributed protocol

TM for Embedded: Wind River Workbench



Symbian phone browser



TM for Enterprise: IBM WebSphere Developer

Remote System Explorer - INDENT2.RPGLE - IBM Rational Software Development Platform

File Edit Source Compile Navigate Search Project Run Window Help

Remote Systems | Team

- New Connection
- Local
- My System i Connection
  - iSeries Objects
    - Work with libraries...
    - Work with objects...
    - Work with members...
    - Library list
    - User libraries
    - compare mbrs
    - qrpglesrc
    - indented members
      - INDENT.rpgle
      - INDENTFREE.rpgle
      - INDENTMAX1.rpgle
      - INDENTMAX2.rpgle
      - INDENT2.rpgle
      - INDENT5.rpgle
      - INDENT6.rpgle
  - iSeries Commands
  - iSeries Jobs

INDENTFREE.RPGLE | INDENT2.RPGLE

Line 70 Column 43 Replace

```
     .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2++++++++++++++++++++
005800       *
005900       * MOD --PROGRAMMER-- MM/DD/YY ------------CHANGE-MADE-----------
006000       * 001 X. Xxx          mm/dd/yy rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
006100       *
006200       *
006300       ****************************************************************
006400       *
006500       * VAR 01 - DOU
006600       *
006700     C                    DOU        a = b
006800     C                    DOU(m)     a = b
006900     C                    DOU   (r)  a = b
007000     C                    DOU        a + b + c
007100     C                               =
007200     C                               d + e + f
007300     C                    DOU   (m)  a * b
007400     C                               =   c *d
```

Properties | Remote Scratchpad

Property	Value
Attribute	SRC
Name	INDENTMAX1

Remote System Det... | iSeries Error List | Remote Shell | iSeries Table View | iSeries Commands... | Navigator | Tasks

Remote system filter My active jobs

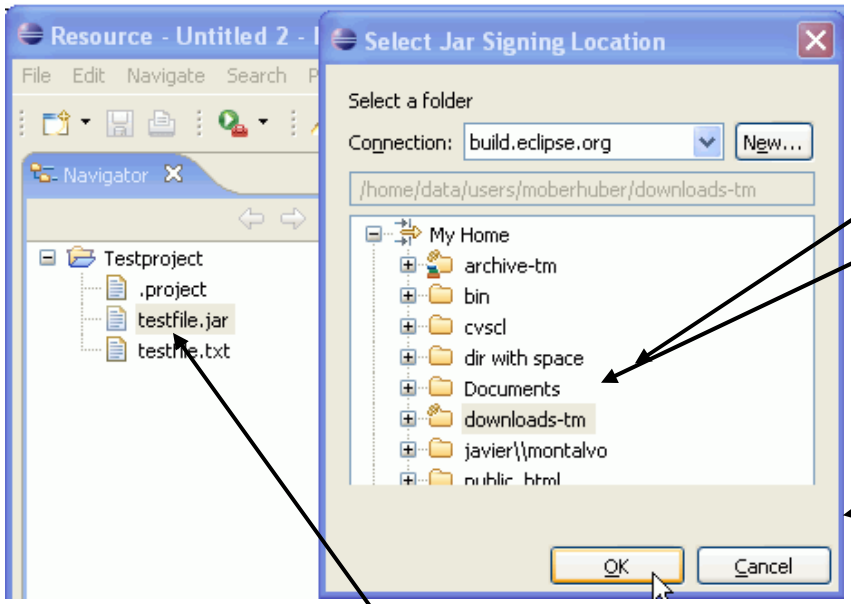
Name	User	Number	Status	Subsyst
145957/QUSER/QZRCSRVS	QUSER	145957	*ACTIVE	QUSRW
145974/QUSER/QJVACMDSRV	QUSER	145974	*ACTIVE	QUSRW

Using RSE APIs (Example: Creating a Connection)

```
public void run() {
    String hostName = "build.eclipse.org"; //$NON-NLS-1$
    ISystemRegistry registry = RSECorePlugin.getDefault().getSystemRegistry();
    ISystemProfile profile = registry.getSystemProfileManager()
        .getDefaultPrivateSystemProfile();
    IHost host = registry.getHost(profile, hostName);
    if (host == null) {
        host = registry.createHost(
            "SSH Only",    //System Type Name
            hostName,      //Connection name
            hostName,      //IP Address
            "Connection to Eclipse build site"); //description
    }
}
```

See the EclipseCon 2007 TM Tutorial for more programming examples

RSE Tools for Remote Files



Model

Subsystem: IRemoteFile, IRemoteFileSubSystem

Service: IHostFile, IFileService

View

SystemRemoteFileDialog

SystemRemoteFolderDialog

Controller

RemoteFileUtility

UniversalFileTransferUtility

WorkspaceResourceSet

RSE brings some re-usable widgets that operate on the generic services and subsystems contributed (any kind of transport can be plugged in).

Why are there Subsystem and Service layers?

- Originally, RSE just dealt with Subsystems
  - You can register just ANYTHING as a Subsystem.
- It turned out, that some **Subsystems should be used with multiple protocols** (e.g. files-via-dstore, files-via-ssh, files-via-ftp)
  - The Service Layer allows to replace the protocol
  - UI code, filters, widgets etc. are re-used from the Subsystem
- The Subsystem is the client-facing side (filters, dialogs, …) although it has both a non-UI layer and a UI layer (via Adapters).
- The Service is always non-UI. It's for programmers.
- For your own subsystem, you can but don't have to do a Service.

Extension Points

- RSE
  - Adding custom system types / wizards
  - Adding custom services / subsystems / filters
  - Adding custom actions
  - Customizing Persistence
- Terminal
  - Adding custom connectors
  - More API work to be done for 3.0 – lots of good suggestions
- Service Discovery
  - Adding custom protocols

RSE: Custom Service for Existing Subsystem

- Goal: Add a new protocol (WebDAV) for using the RSE Remote File Browser on it. Works exactly the same for other protocols
- Tasks:
  - Have a generic **Service** for WebDAV (independent of RSE). Write an IFileService wrapper for it, using IHostFile objects as model.
  - Register the **subsystemConfigurations** extension point. Re-use FileServiceSubsystem, but adding the plumbing for an FTP ConnectorService.
  - Write an WebDAVFileAdapter, and register an AdapterFactory for it in the Activator.
- That's simple, because IFileService is simple!

RSE: Fully Custom Subsystem

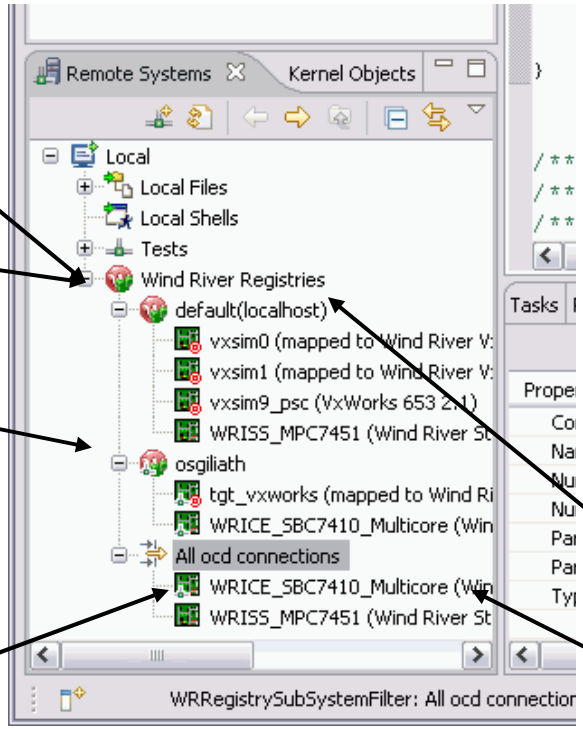
Model

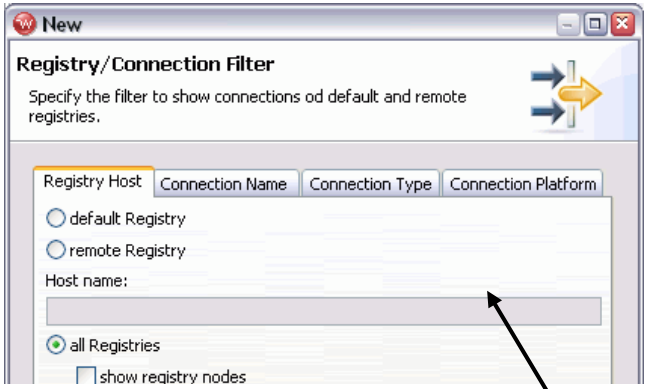
ISubSystemConfiguration
Provides special filtering

ISubSystem

ISystemFilterReference

Original Model Objects
from previous generation of product





SystemFilterStringEditPane
User-friendly filter editng

ISubSystemConfigurationAdapter
Presentation of Subsystem, Options for Filters

ISystemViewElementAdapter
Adapts them to RSE

View

TM Future Planning

TM 3.0 Plans

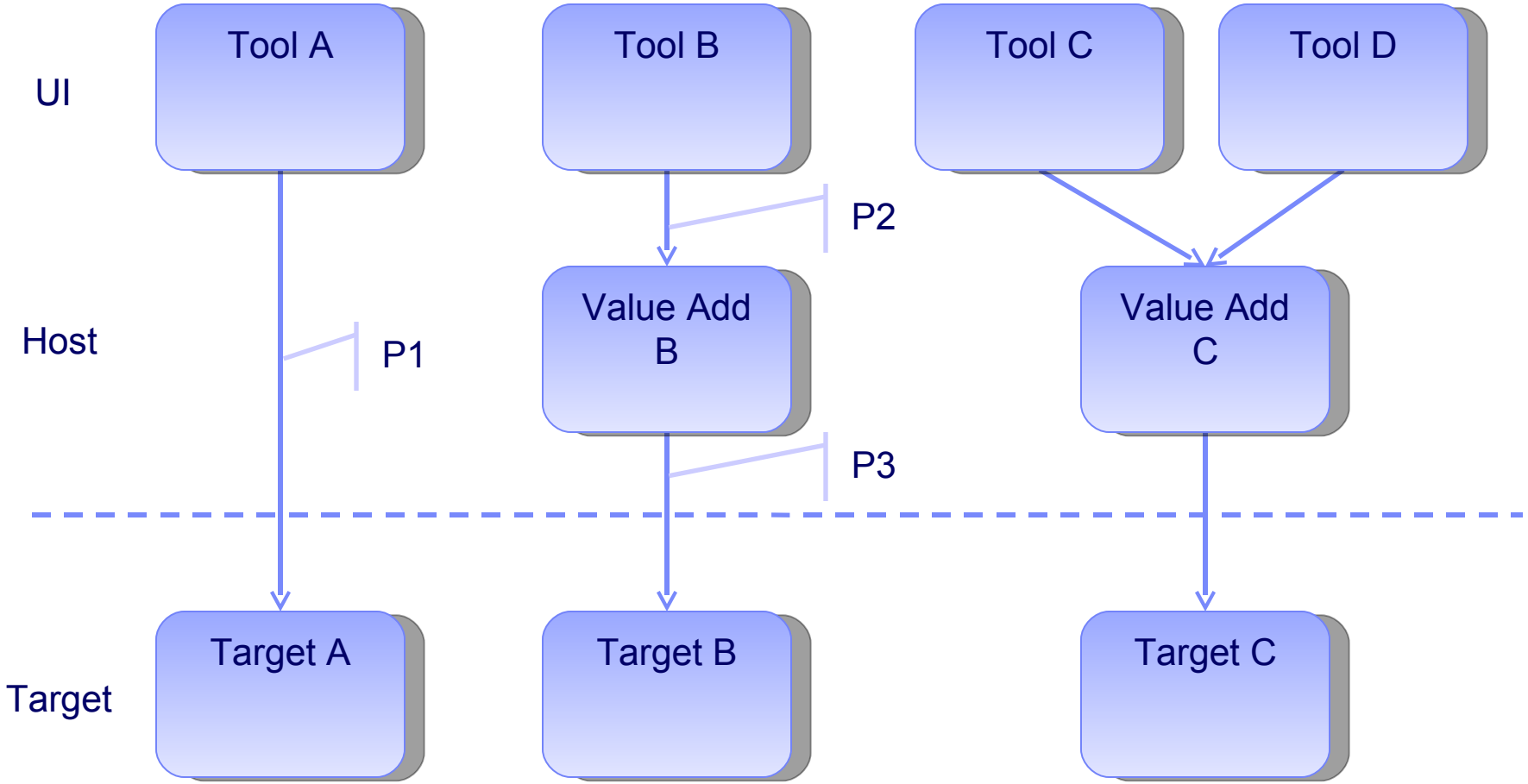
- Going to join the Ganymede release train
  - TM 3.0 release in June 08 will contain some API changes

- A preliminary collection of potential plan items has been collected on the Wiki at http://wiki.eclipse.org/index.php/TM_Future_Planning
  - **Quality** – Reduce bug backlog, improve performance, API cleanup & hardening
  - **Scaling Down** – Further UI/Non-UI splitting, componentization, becoming more RCP-aware and applicable for headless
  - **Migration** and Import/Export of connections (Persistence)
  - Improved **Remote CDT** Integration
  - **Target Connection Framework (TCF)**
  - Further collaboration with other Eclipse projects – **needs YOU!**
    - Google SoC WebDAV; Platform/Team Synchronization
    - TPTP; ECF; EFS improvements
    - SWT deferred download on drag&drop (bug 196176)

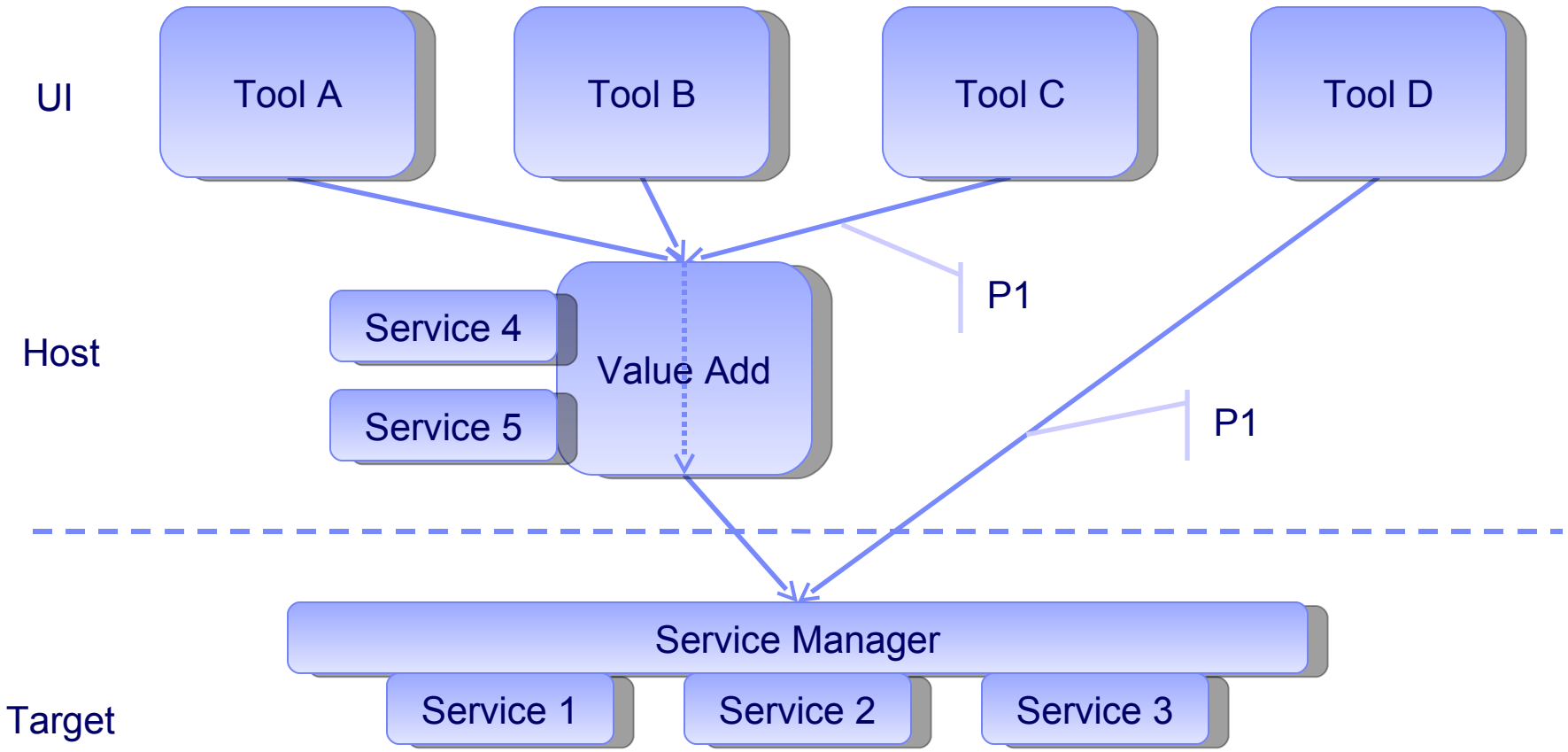
Target Communication Framework (TCF)

- Today almost every device software development tool on the market has its own method of communication with target system.
  - Individual setup for each communication method
  - Especially awkward for multi-core (different tool for each core)
- The goal is a single protocol used to communicate between all tools and targets, supporting auto-discovery, multiplexing and tunneling
  - Transport protocol agnostic
  - Single point of configuration, single link

TCF: Example of Existing Architectures

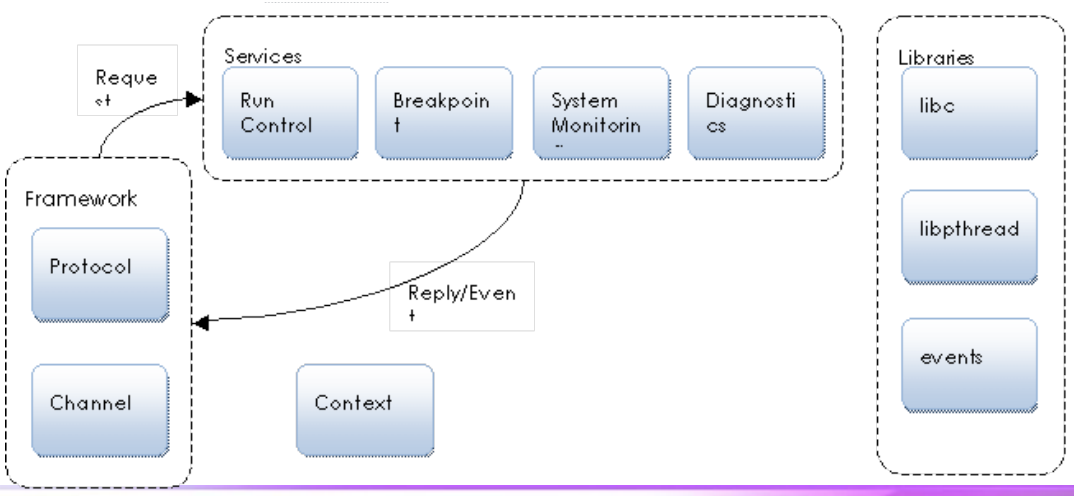


TCF: Vision



TCF Lightweight Open Agent

- TCF is mostly about the Protocol
- But there is a lightweight plain C extendable agent, with some sample implementations:
  - Linux ptrace basic debugging
  - Linux remote file service (integrated with RSE)
  - Linux remote process service (integrated with RSE)



Mission, Goals and Future

- **TM Mission:** *Create data models and frameworks to configure and manage remote systems, their connections, and their services.*
- **TM Vision:** *To be the Eclipse "Explorer of Network Neighborhood", supporting interactive discovery and drill-down to remote services, and passing context info to higher-level components.*
- **Ideas being discussed (Bugzilla Items)**
  - Abstract Descriptions of Remote Systems and Services (for Search, integration with project and build); with DD-SPIRIT / IP-XACT group
  - Dynamic addition of Services, further improved Auto-Discovery
  - Multi-core / Multi-target support through connection groups
  - Adapters for Target access control (shared board labs)
  - Component-Based Launching (CBL)
- See the TM Wiki, and the TM Use Cases Document
  http://www.eclipse.org/dsdp/tm/doc/DSDPTM_Use_Cases_v1.1c.pdf

DSDP: Device Software Development Platform

Goal: Develop framework and exemplary tools for device software development .

"Traditional" Software Dev.	Device Software Dev.
Local (changing!)	Remote
Hardware abstraction through OS	Custom hardware (DSP, Multi-core, FPGA, …)
Standards based	Many vendors – many connection schemes
	Managed targets / access restrictions
	Secure tunnels for deployed targets

Device Software Vendors want to plug-in solutions for their specific area of expertise → need a highly modular, pluggable framework for connectivity

TM Mission in Detail

- Target Management Project
  - creates **data models** and **frameworks**
    - Abstraction of remote system properties
    - Meta-framework for services to plug in and discover
  - to **configure** and **manage**
    - Targets, processors, cores, scan chains, processes, threads
    - Configuration information of deployed systems, Kernel version etc.
    - Board labs, access restrictions, credentials
  - their **connections**,
    - TCP, UDP, Serial, various agents, ICE boxes, …
    - Tunnelling to deployed targets
  - and their **services**.
    - Download, run, debug, query information, reset, FLASH utility, …
    - UI for configuring services, and UI-less delegate for execution

TM: Service-based approach

Typical services to be provided include

- OS-Aware Services
  - Remote File System (Plus mappings for cross-mounted file systems)
  - Remote Process Explorer (Query target info, Kernel Objects – more abstraction)
  - Kernel Module Downloads
- Services that do not necessarily need an OS
  - Remote Console (Serial, TCP/IP, …)
  - Reset / Reboot (Start and Stop Cores)
  - RAM download of arbitrary images (e.g. via JTAG)
  - FLASH utility
  - Debugger Launcher

- Services provide "scriptable actions" → Component Based Launching
  - Need pluggable services to be contributed by vendors
  - Want to **autodetect services** as much as possible

Eclipse as the Integration Platform

- Today, embedded software vendors choose Eclipse because it's a nice toolkit for their own stuff.
- Tomorrow, we can work on Open Standards and Integrations
  - Select board A from vendor B
  - Have the build system automatically choose the compiler (from vendor X)
  - Have the debugger automatically choose the ICE setup (from vendor Y)
  - Have TOS awareness automatically configured (from vendor Z)
- For Open Standards we need to collaborate – it's YOUR opportunity

Remote Development, where are you going?

- Embedded (DSDP) typically does cross development
  - Local workspace (files) and compile
  - Run and debug with a lightweight remote agent
- Remote Development needs similar tools (remote file, remote shell access) but has different focus and requirements
  - Remote workspace (files) and compile
  - Run and debug with a heavyweight remote runtime / debugger
  - Remote parsing
- We came together because there is much overlap; but there are also discussions about a separate Eclipse Project for “remote development”

Resources, Questions and Answers

- Resources and Pointers
  - TM Homepage, Getting Started, FAQ, Newsgroup, Mailinglist
  - EclipseCon07 Tutorial for Extenders
  - RSE Online User and ISV Docs, Tutorial and Examples
  - Developer Resources: CVS Team Project Sets, TM Bug Process with many good queries, Committer HOWTO,

- Q & A