

OpenROAD 6.2

New Client-Server and Multi-Tier Deployment

New in OpenROAD 6.2 – For all users of OpenROAD.

See: http://community.actian.com/wiki/LoadnRun_Home

Durwin Wright



Disclaimer



This document is for informational purposes only and is subject to change at any time without notice. The information in this document is proprietary to Actian and no part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of Actian.

This document is not intended to be binding upon Actian to any particular course of business, pricing, product strategy, and/or development. Actian assumes no responsibility for errors or omissions in this document. Actian shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. Actian does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

Scope of this presentation



- This presentation contains specific references to Loadrun 6.2
- The information presented is directly applicable to the following
 - Loadrun 5.1
 - Loadrun 6.0
- This presentation can be used to support any of the above versions
- They only differ in the version of OpenROAD that each supports. Other than that they are virtually identical in behavior.
- We recommend the following
 - Use Loadrun 6.2 for the Loadrun 6.2 Server
 - Use Loadrun 5.1, Loadrun 6.0 or Loadrun 6.2 on the clients
- Loadrun 6.2, Loadrun 6.0 and Loadrun 5.1 are available now!

Contents

➔ OpenROAD 6.2 -New Client-Server and Multi-Tier Deployment

- Features illustrated in the presentation will require the first OpenROAD 6.2 patch
 - p14746 or later
 - Loadnrun 6.2 Installer
 - Loadnrun 6.0 Installer
 - Loadnrun 5.1 installer
- This is the second of three presentations covering OpenROAD 6.2



- See: http://community.actian.com/wiki/LoadnRun_Home

Meeting OpenROAD 6.2 Objectives ...



Improve the deployment of OpenROAD applications

→ From the customer viewpoint:

- Provide a comprehensive and straightforward mechanism for deploying OpenROAD
- Provide richer and more robust support for management of OpenROAD Client-Server and OpenROAD Multitier architectures
- Make migrations to new versions of applications running the same version of OpenROAD easier
- Make stepwise redeployment of existing Client-Server OpenROAD applications as OpenROAD Multitier applications a feasible process
 - Currently this redeployment requires all the SQL processing throughout all the client applications to be moved into the Server application before anything can be released
 - Clients can view this as too big and risky a project.

Application deployment using Loadrun

- 
- Loadrun provides an infrastructure for OpenROAD deployment
 - Client-Server based
 - Application Server based
 - The aim of the infrastructure is to minimize the deployment effort and risk, and to maximize the range and load that can be handled
 - Central administration
 - Deployment of latest versions is automatic
 - Existing client batch processing is readily integrated into Loadrun
 - Simple-to-use client management tools (where appropriate)
 - The same client machine can run multiple OpenROAD versions (6.2, 6.0, 5.1), multiple installations, multiple applications, multiple instances, multiple users ...
 - Loadrun design was guided by client needs and client experience

Loadnrun Benefits (already a reality)

→ We have already seen practical benefits from the Loadnrun facility:

■ A client has already used OpenROAD 6.2 Loadnrun to deploy and manage their environment:

- Deployed to numerous PCs
- 5.1.1 and 6.0.2 (and testing OR 6.2.0) OpenROAD applications using the same client machine
- Very few problems

■ Actian uses it internally to:

- To access and compare older versions of our applications when regression testing
- To test and expose new features under development and compare behaviours of previous versions of OpenROAD side-by-side!

Easier partitioning using optional Ingres Net



- Direct connection from Multitier Client to database
 - Instead of / as well as the OpenROAD Server connection
- Loadrun Launcher uses the local Ingres Net Client configured by Loadrun
- The Query Tool provided with the Loadrun installation includes a vnode-definition wizard that configures the client
- The Loadrun demos show how to use dynamic vnodes that do not require any pre-defined vnodes
- Examples are also provided that show how to create a vnode when the application is first setup after Loadrun downloads it

Original Loadrun

- 
- Loadrun was jointly developed by Actian and Customers
 - Original Objectives
 - Package eClient runtime as an MSI Installer
 - Deployment of Loadrun eClient applications will not require any special privileges for the end user
 - Customer Contribution
 - Loadrun Client and Server Application
 - Visual Studio-based Setup Bootstrapper
 - Original documentation on Community Wiki
 - Actian Contribution
 - OpenROAD eClient runtime (based on OR 5.0)
 - Originally provided irunnerw.exe (was later deprecated in favor of w4glapp.exe)
 - Developed w4glapp.exe (fused version of w4gldev.exe)

New Loadrun (Part 1 of 3)



- Support for concurrent versions of OpenROAD
 - Loadrun Client 5.1
 - Loadrun Client 6.0
 - Loadrun Client 6.2

- Allow host of any version of OpenROAD eClient by Loadrun Server
 - Loadrun Server 5.1 (based on OpenROAD 5.1)
 - Loadrun Server 6.0 (based on OpenROAD 6.0)
 - Loadrun Server 6.2 (based on OpenROAD 6.2)

- Packaged optional Standalone Net Client
 - Based on Ingres 10.1.1 Net Client (or 10.0.0 for Loadrun 5.1/6.0)
 - Hardcoded Installation Codes (XN, XO, XP)

New Loadnrun (Part 2 of 3)

- 
- Packaged as WiX MSI project
 - Provided 4GL-based bootstrapper
 - Allow Silent or Reduced UI install option
 - Support installation into latest versions of Windows with or without UAC enabled
 - Windows 7 and Windows Server 2008 R2
 - Windows 8, Windows 8.1, Windows 10.0 and Windows Server 2012
 - Incorporated all customer enhancement requests
 - Loadnrun Server Host Isolation
 - Loadnrun Compression
 - PRERUN and POSTRUN scripts
 - Provide feedback during download of user applications
 - Log all Loadnrun Client access to Loadnrun Server

New Loadnrun Objectives (Part 3 of 3)

→ Provide Several Types of Demos

- AppServer-based demos
- Standalone demos
- Two-tier Net Client demos

→ Provide Simple Launcher

- Demonstrates how to launch a typical application
- This is just a demo but can be incorporated and modified as each customer sees fit

→ Provide source code for all 4GL applications

→ Updated Wiki Documentation

http://community.actian.com/wiki/LoadnRun_Home

→ Add Loadnrun to OpenROAD Documentation set

Getting Started with Loadrun Client



Loadnrun Terminology

→ Server

- Loadnrun Gatekeeper
- Loadnrun Server
- Loadnrun Server Runtime
- Loadnrun Server eClient Host Directory

→ Client

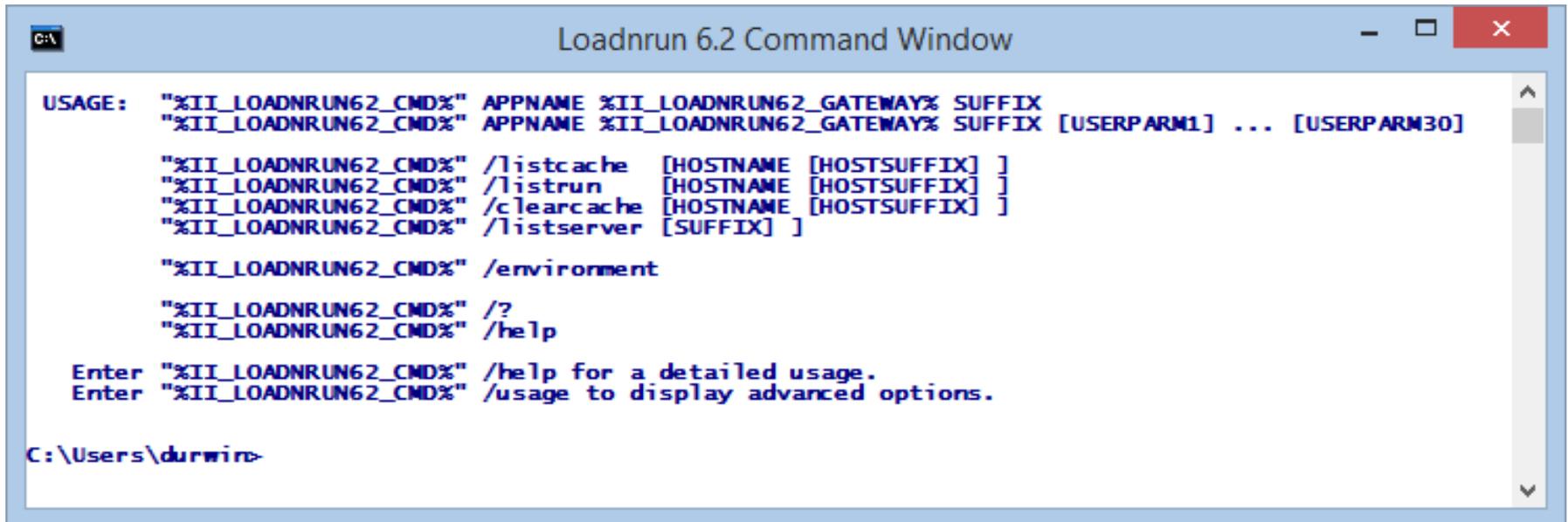
- Loadnrun Client Runtime
- Loadnrun APPNAME
- Loadnrun GATEWAY URL
- Loadnrun SUFFIX
- Loadnrun HOSTNAME
- Loadnrun eClientCache

Loadnrun APPNAME Launching

- 
- The shortcuts for Loadnrun 6.2 are located under the following
Start→All Programs→Actian Loadnrun 6.2
 - The shortcut to launch the Loadnrun Command Window is
Start→All Programs→Actian Loadnrun 6.2→Resources→Loadnrun 6.2 Command Window
 - These shortcuts will only appear under the Administrative account that was used to install the Loadnrun client runtime
 - The Loadnrun Command, “%II_LOADNRUN62_CMD%”, can be used by any user in a command window or a shortcut
 - Essentially, once the Loadnrun Client runtime is installed, the only thing needed on the client machine to deploy and application is the appropriate shortcut

"%II_LOADNRUN62_CMD%"

- The output of the command, "%II_LOADNRUN62_CMD%", is shown in the frame below
- This command can be used to launch Loadrun applications and manage the local cache of OpenROAD Applications



```
C:\Loadrun 6.2 Command Window

USAGE: "%II_LOADNRUN62_CMD%" APPNAME %II_LOADNRUN62_GATEWAY% SUFFIX
"%II_LOADNRUN62_CMD%" APPNAME %II_LOADNRUN62_GATEWAY% SUFFIX [USERPARAM1] ... [USERPARAM30]

"%II_LOADNRUN62_CMD%" /listcache [HOSTNAME [HOSTSUFFIX] ]
"%II_LOADNRUN62_CMD%" /listrun [HOSTNAME [HOSTSUFFIX] ]
"%II_LOADNRUN62_CMD%" /clearcache [HOSTNAME [HOSTSUFFIX] ]
"%II_LOADNRUN62_CMD%" /listserver [SUFFIX] ]

"%II_LOADNRUN62_CMD%" /environment

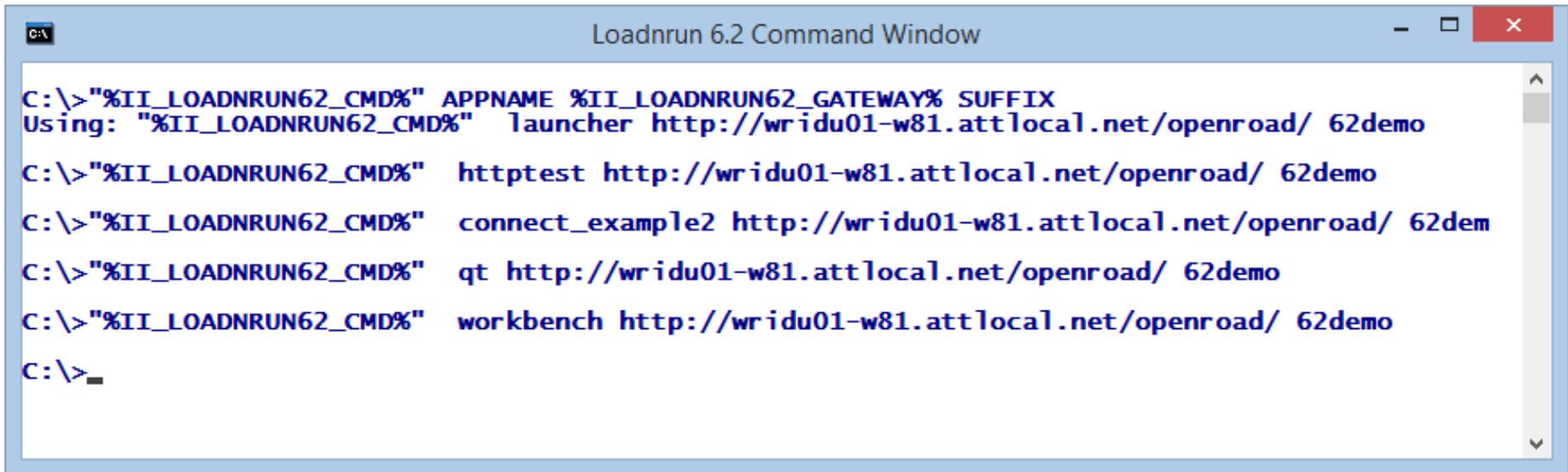
"%II_LOADNRUN62_CMD%" /?
"%II_LOADNRUN62_CMD%" /help

Enter "%II_LOADNRUN62_CMD%" /help for a detailed usage.
Enter "%II_LOADNRUN62_CMD%" /usage to display advanced options.

C:\Users\durwin>
```

Start...

- ...Loadnrun Simple Launcher
- ...Loadnrun httpptest
- ...Loadnrun connect_example2
- ...Loadnrun qt
- ...Loadnrun workbench



```
C:\>"%II_LOADNRUN62_CMD%" APPNAME %II_LOADNRUN62_GATEWAY% SUFFIX
Using: "%II_LOADNRUN62_CMD%" launcher http://wridu01-w81.attlocal.net/openroad/ 62demo
C:\>"%II_LOADNRUN62_CMD%" httpptest http://wridu01-w81.attlocal.net/openroad/ 62demo
C:\>"%II_LOADNRUN62_CMD%" connect_example2 http://wridu01-w81.attlocal.net/openroad/ 62dem
C:\>"%II_LOADNRUN62_CMD%" qt http://wridu01-w81.attlocal.net/openroad/ 62demo
C:\>"%II_LOADNRUN62_CMD%" workbench http://wridu01-w81.attlocal.net/openroad/ 62demo
C:\>
```

Loadnrun Simple Launcher

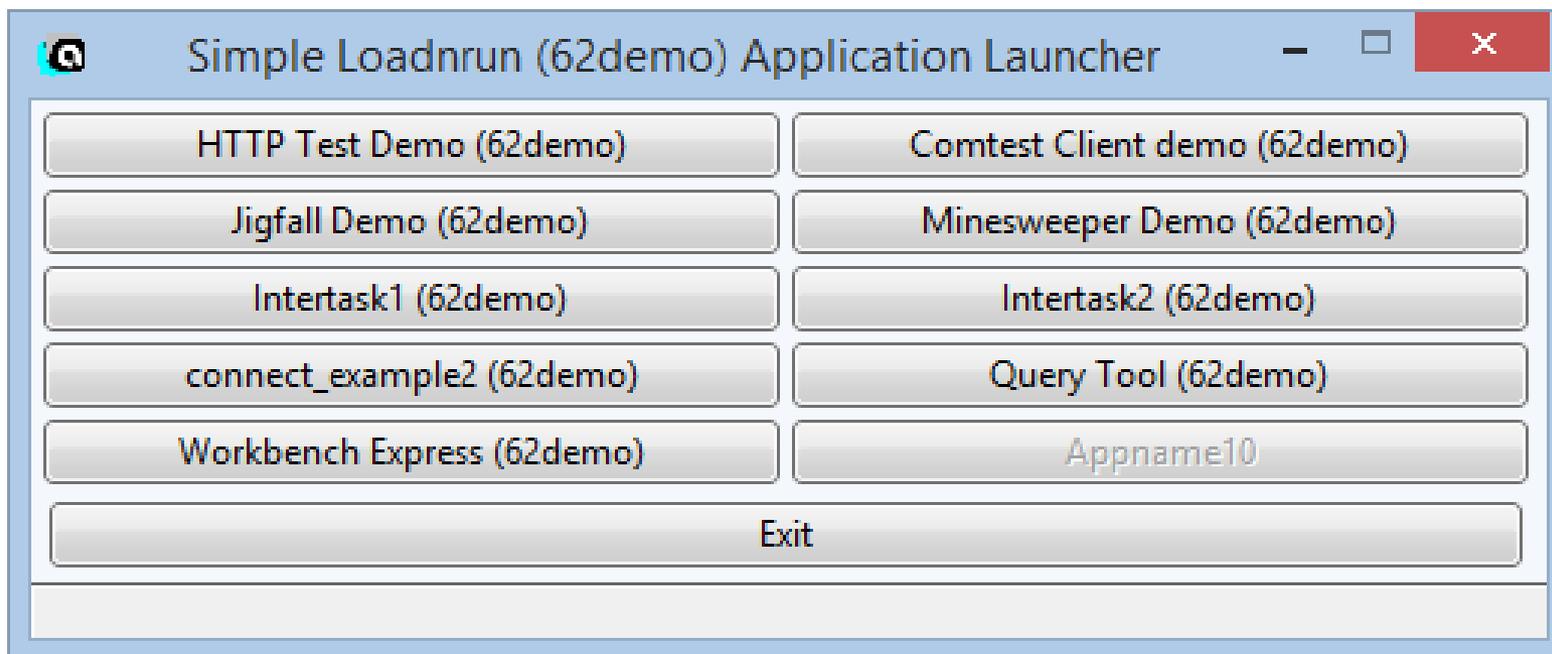


Loadnrun Simple Launch

- Simple application to launch Loadnrun applications
- Sample applications hosted on Loadnrun Server under the 62demo suffix

This can be launched via a shortcut

- Actian Loadnrun 6.2 → Launcher



HTTP Test Demo (62demo)



The Simple Launcher can launch this application

Notice the location URL is configured automatically

Example of AppServer Application

Provided as a demo with Loadnrun

A screenshot of a Windows application window titled "HTTP Test Windows". The window contains two sections of input fields. The first section, "RemoteServer Initiate Parameters", includes fields for "image" (value: comtest), "flags" (value: :::gooduser:goodpw), "location" (value: http://wridu01-w81.attlocal.net), and "routing" (value: http). The second section, "RemoteServer Call4GL Parameters", includes fields for "procname" (value: helloworld), "hellostring" (value: HTTP Test Loadnrun!), and "counter" (value: 1). At the bottom of the window are three buttons: "Initiate", "Call4GL", and "Release".

RemoteServer Initiate Parameters	
image	comtest
flags	:::gooduser:goodpw
location	http://wridu01-w81.attlocal.net
routing	http

RemoteServer Call4GL Parameters	
procname	helloworld
hellostring	HTTP Test Loadnrun!
counter	1

Buttons: Initiate, Call4GL, Release

Connect Example 2 (62demo)

The Simple Launcher can launch this application

This application can be used to generate a dynamic vnode

Example of Two-Tier Client Server application using the S/A Net Client

Example of use of “Dynamic Vnode”

Provided as a demo with Loadnrun

Dynamic Vnode Connection Demo

connection string: iibdodb/ingres

vnode: vnode

addr: localhost tcp_ip ll

attribute: Direct Connect

user: username

password:

database: iibdodb

server: ingres

EP_INTERACTIVE DEBUG Enabled

Query Tool (62demo)

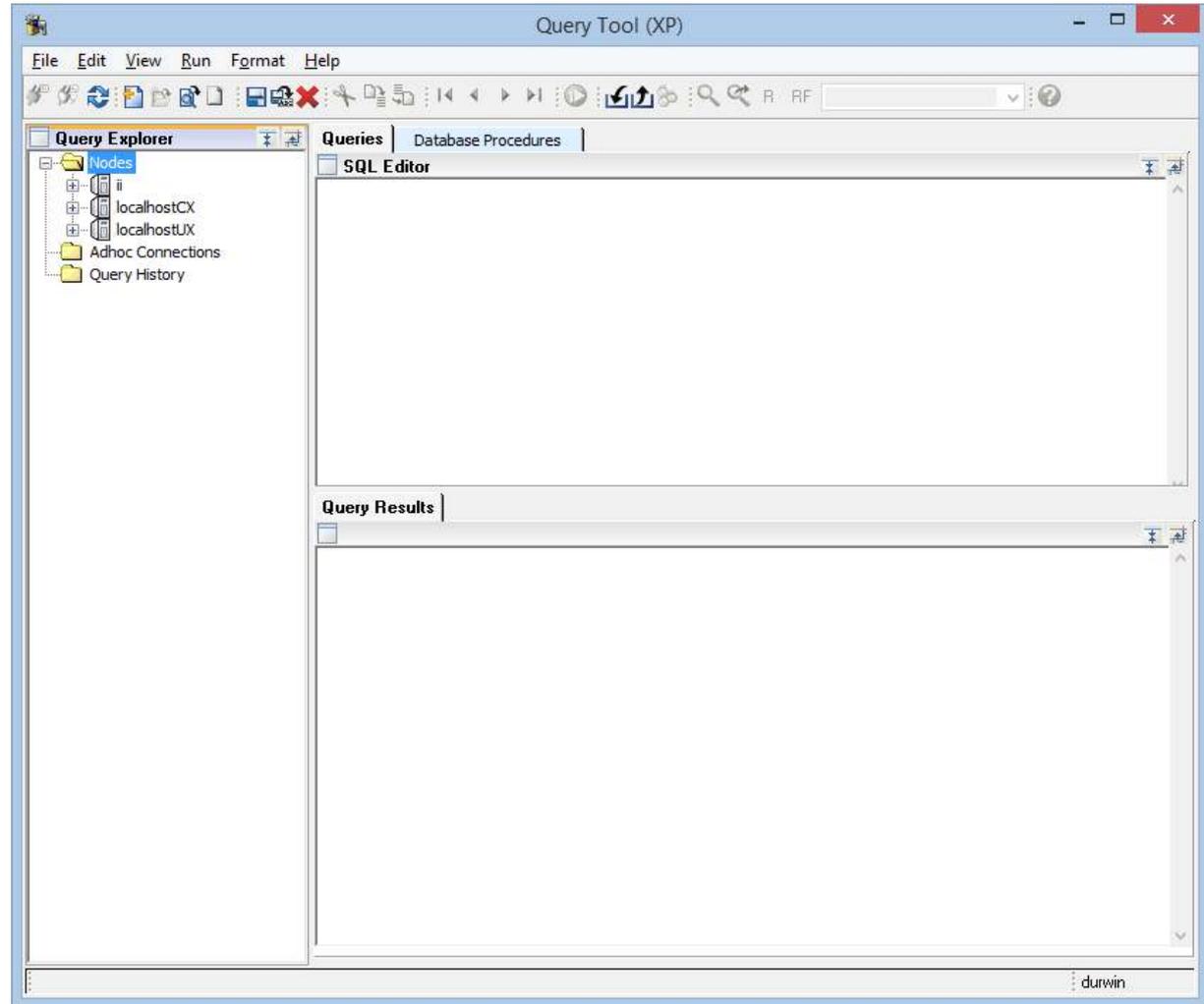


The Simple Launcher can launch this application

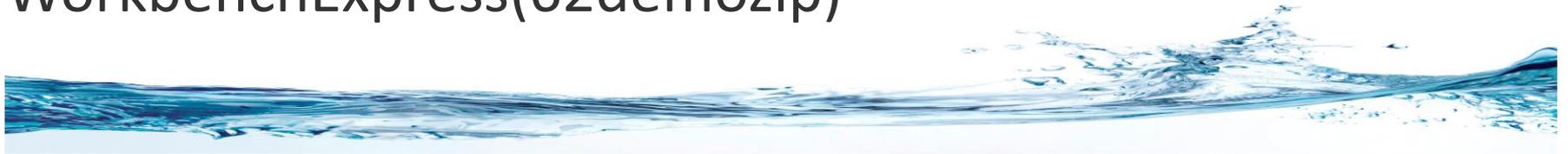
This is the Query Tool application

Example of more sophisticated Client/Server application

Provided as a demo with Loadrun



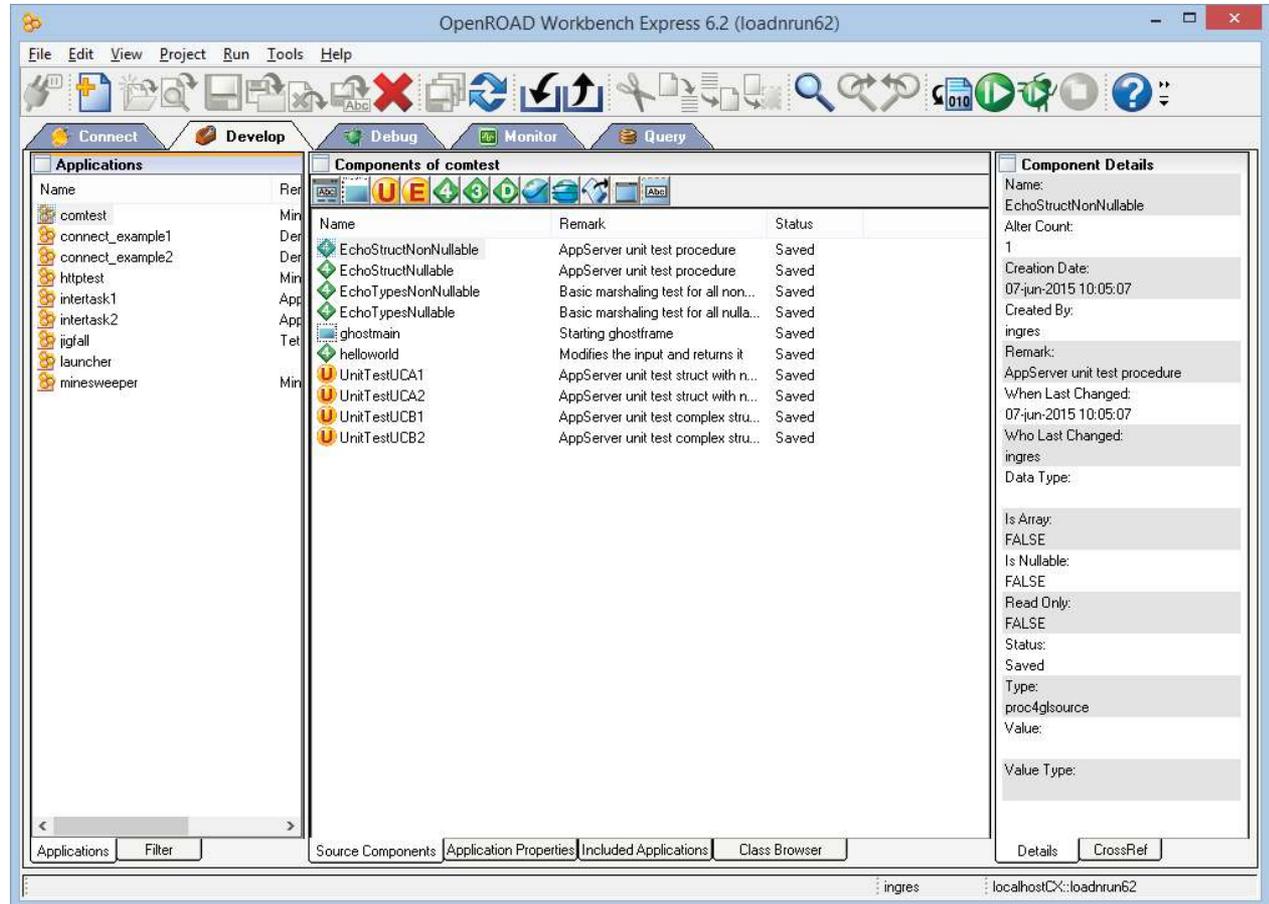
WorkbenchExpress(62demozip)



This is the Workbench Express application

Example of an even more sophisticated Client/Server application

Not provided as a demo with Loadrun



Loadnrun Overview



How does Loadrun Work?



- Loadrun Server
 - Loadrun Client
- A single MSI-based installer installs and configures the Loadrun Server and the Loadrun Client
- Loadrun Server hosts the User applications and delivers them to the client machines upon request
- Loadrun Client launches the user application from the Loadrun Server user application location or the Loadrun Client local cache

Loadnrun Generic Features



- All versions of the Loadnrun Client can co-exist on the same machine
- Any version of the Loadnrun Client can launch any version of Loadnrun user applications
- Any version of the Loadnrun Server can host any version of the Loadnrun files
- It is possible to install the Loadnrun Client silently or passively
- Once the Loadnrun Client is installed, all that is required to launch an application is the creation of a simple shortcut or use of a command line
- Effectively the user application installer is reduced to a script that can create the appropriate shortcut on the client machine

Loadnrun Server Features



→ Loadnrun Server Features

- Each Loadnrun application has an install4gl.txt file and any resources (images and files) that are needed
- The images and files can be placed in a compressed archive (typically a zip file)
- The images can also be hosted on a network URL
- A combination of the above is allowed
- The install4gl.txt file is the only file required to host a user application

→ The installation and management of the Loadnrun Server requires Administrative Privileges

Loadrun Client Features



→ Loadrun Client Features

- Download of new versions of a Loadrun applications is automatic
- Version string can be used to force download of new version from server
- Compressed files or network-based image files can be used
- Simple management of applications on local cache
- Simple diagnostic capabilities are built into the product
- Has an optional Ingres Net Client for client/server applications
- Applications from different Loadrun URLs do not share the same eclientcache sub-directory
- Control can be given to a user written script before the Loadrun 4GL application is launched
- Does not require any special privileges to download, launch and run a Loadrun application

→ Installation of Loadrun Client Runtime does require Administrative privileges

Launching Application: Cache Hit on Client



Loadrun Launch – Use Local eclient cache version



User

Loadrun
Client

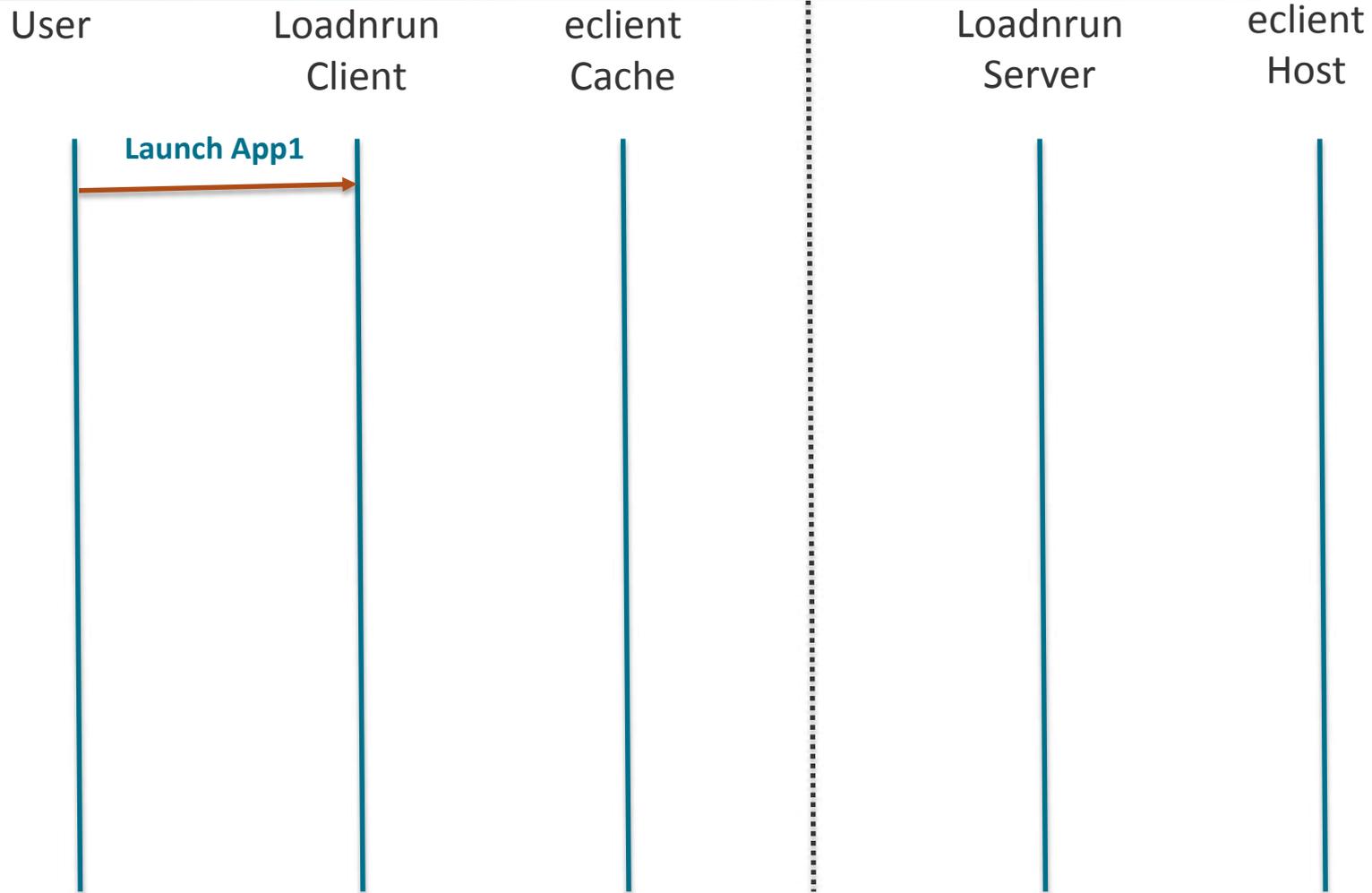
eclient
Cache

Loadrun
Server

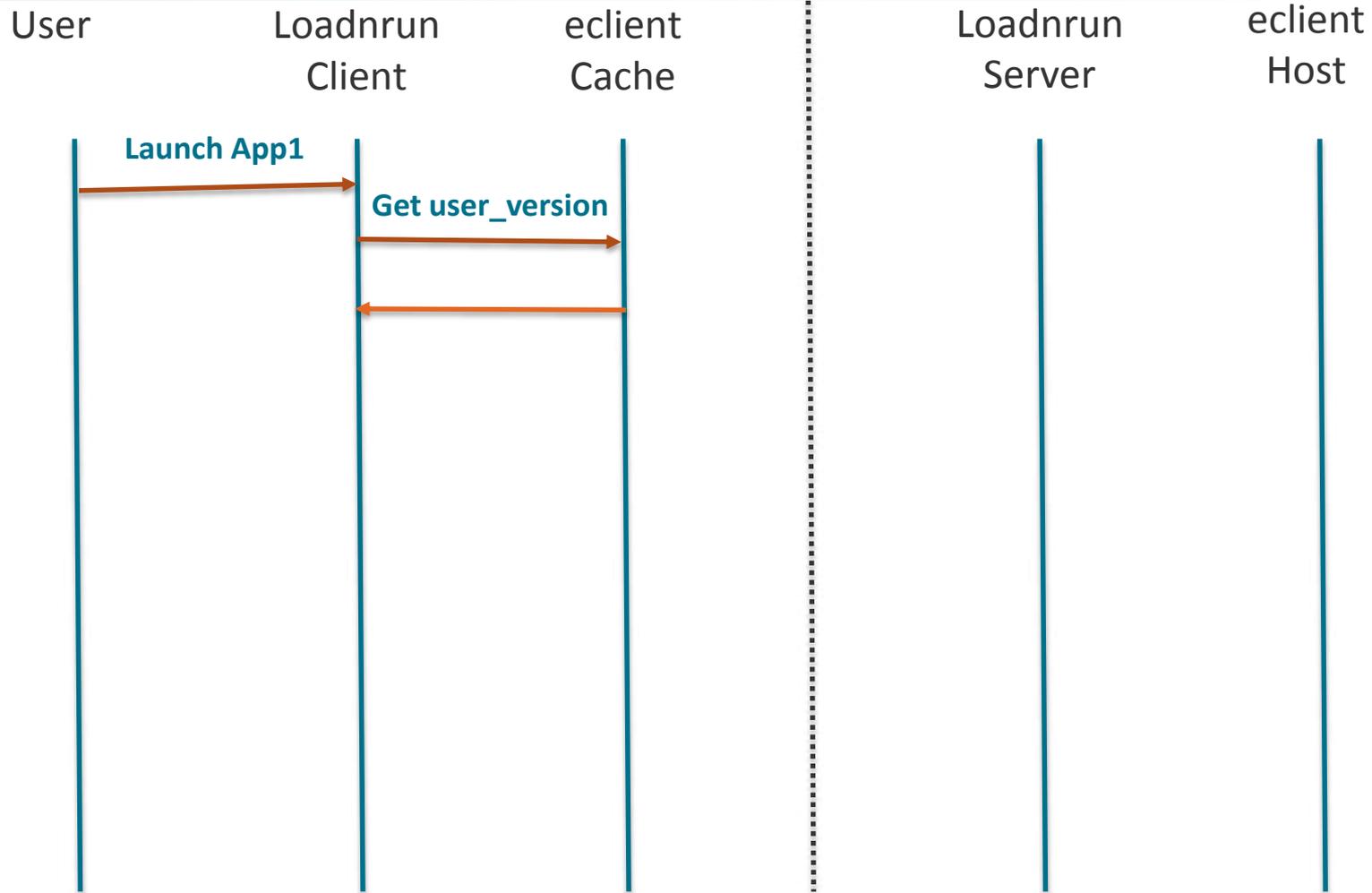
eclient
Host



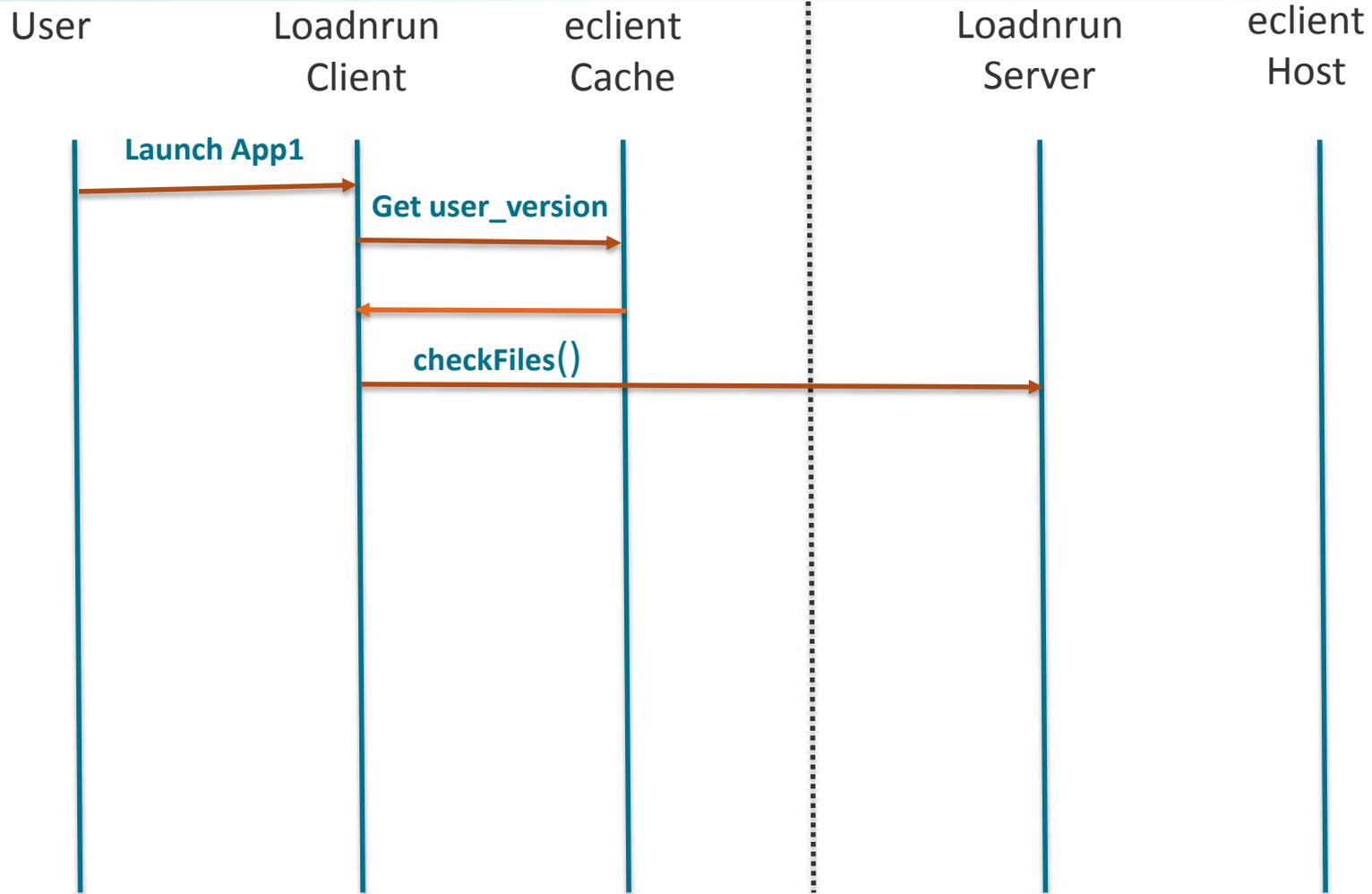
Loadrun Launch – Use Local eclient cache version



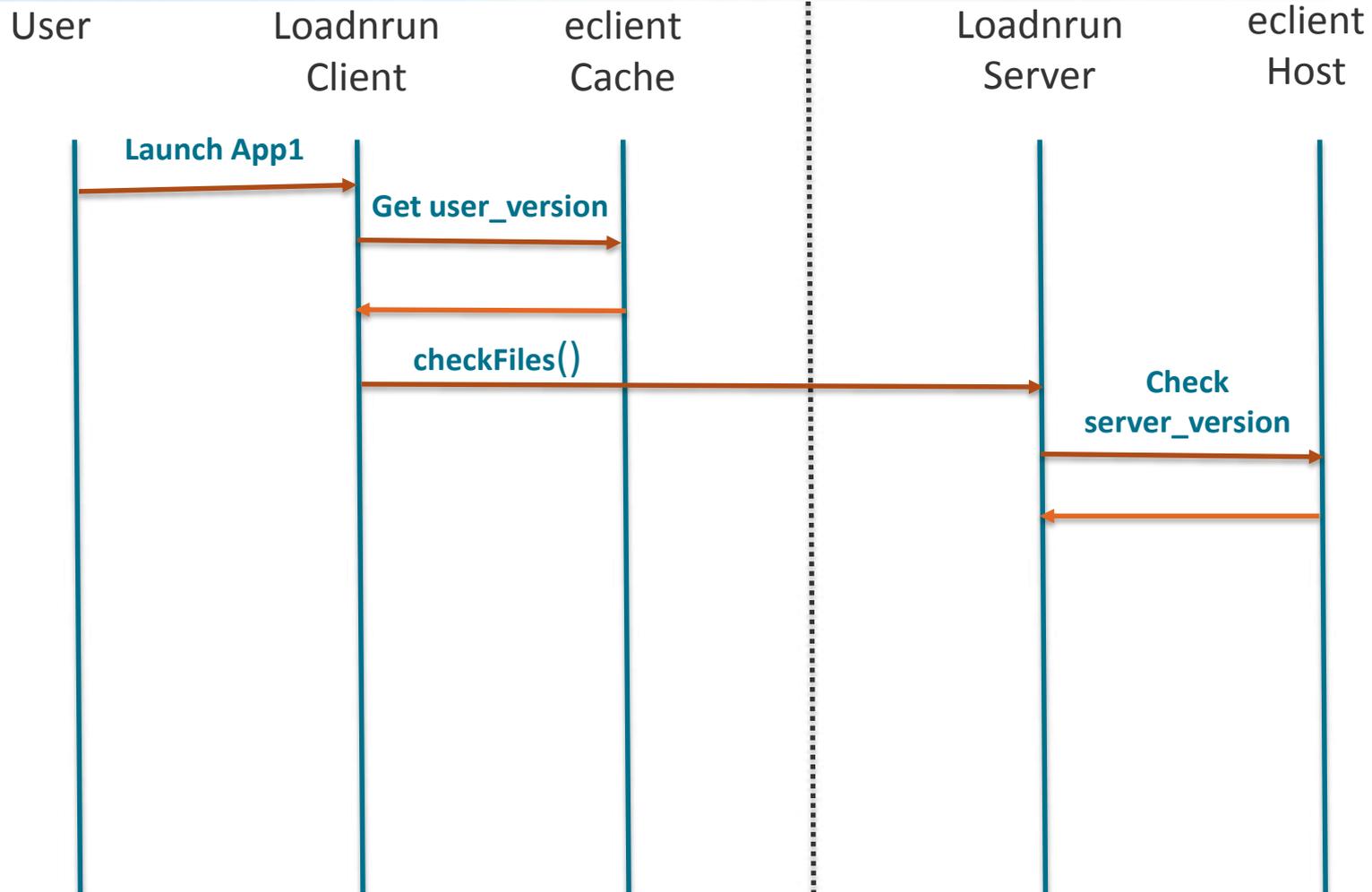
Loadrun Launch – Use Local eclient cache version



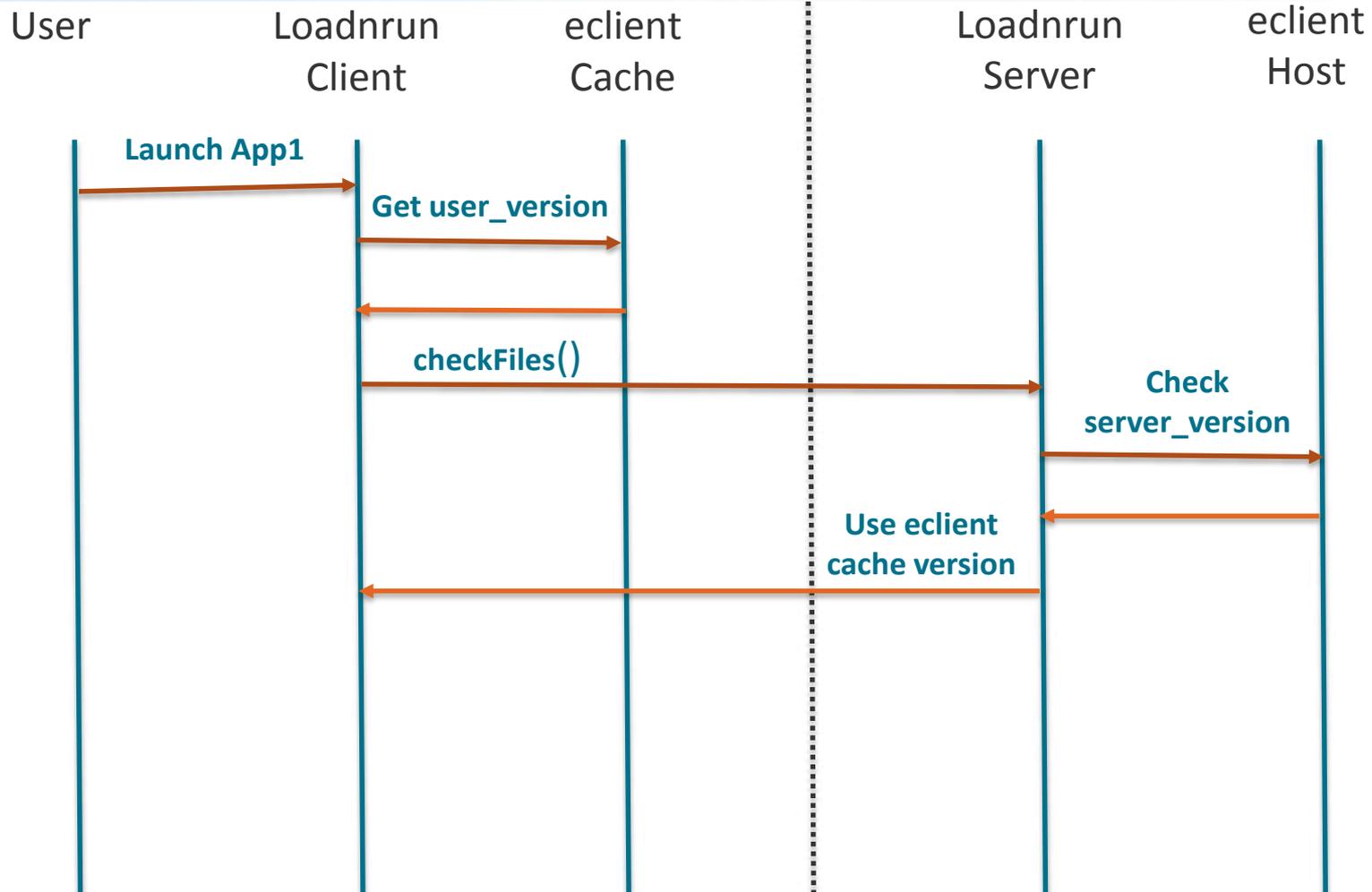
Loadrun Launch – Use Local eclient cache version



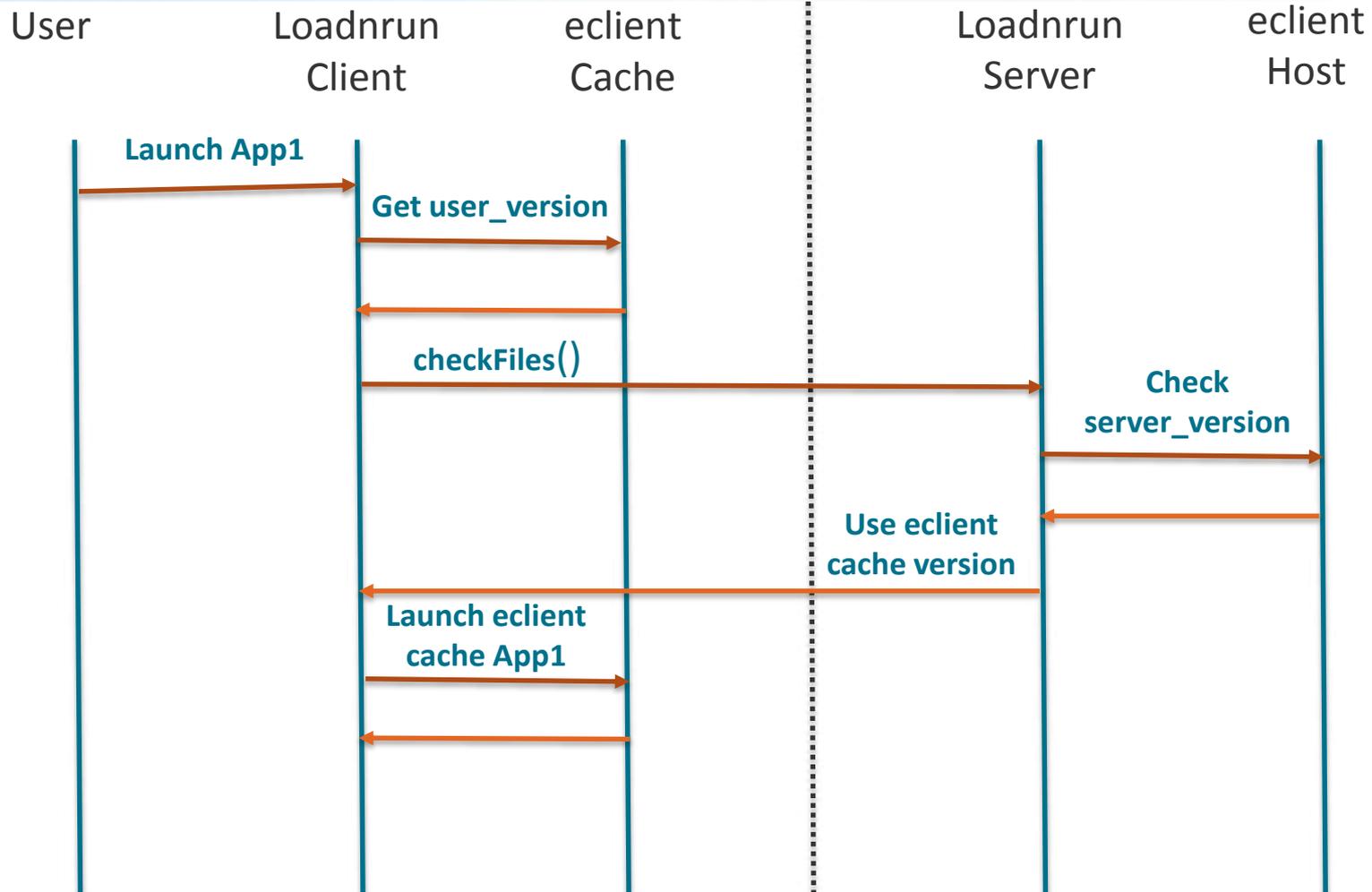
Loadrun Launch – Use Local eclient cache version



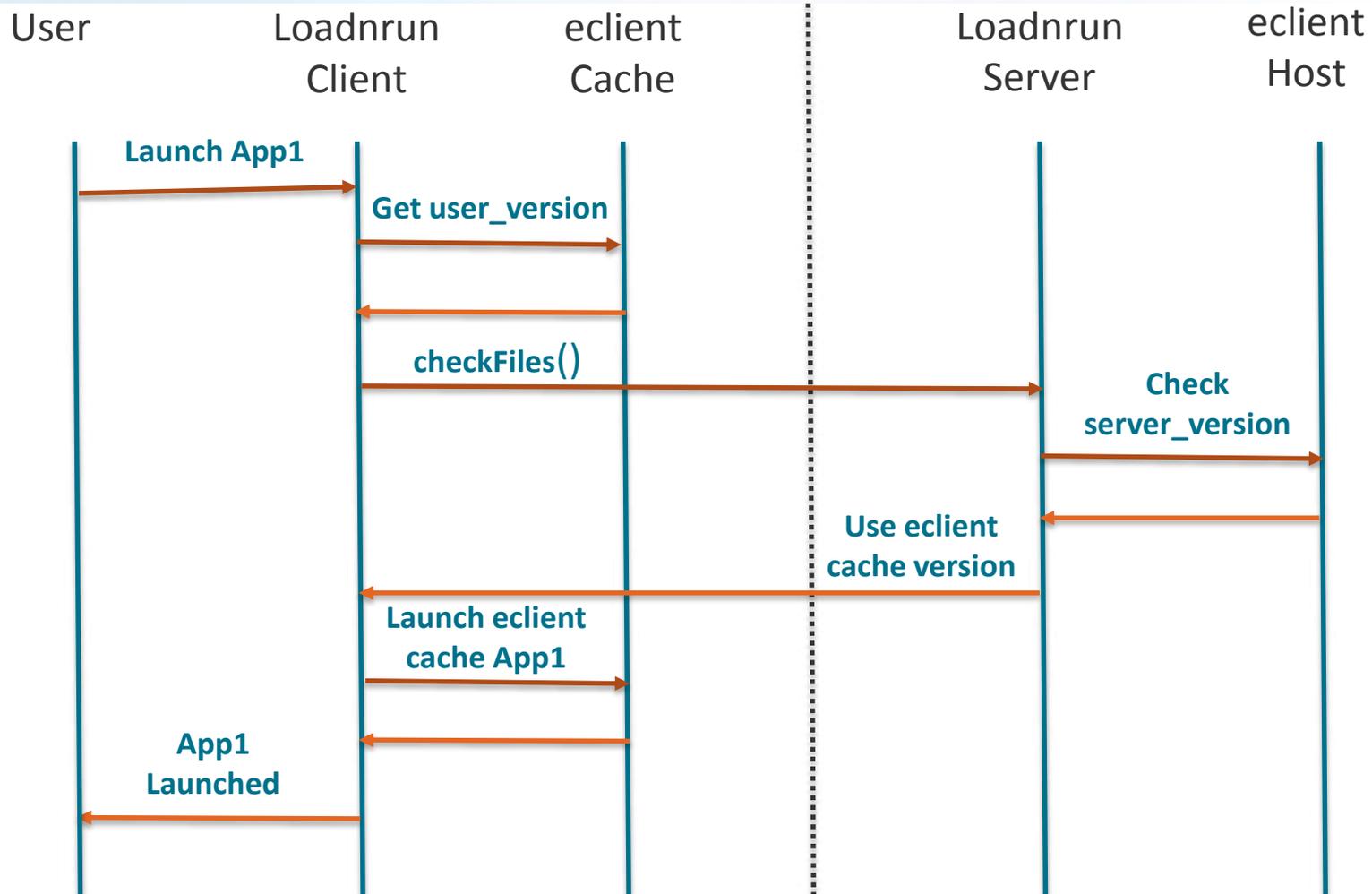
Loadrun Launch – Use Local eclient cache version



Loadrun Launch – Use Local eclient cache version



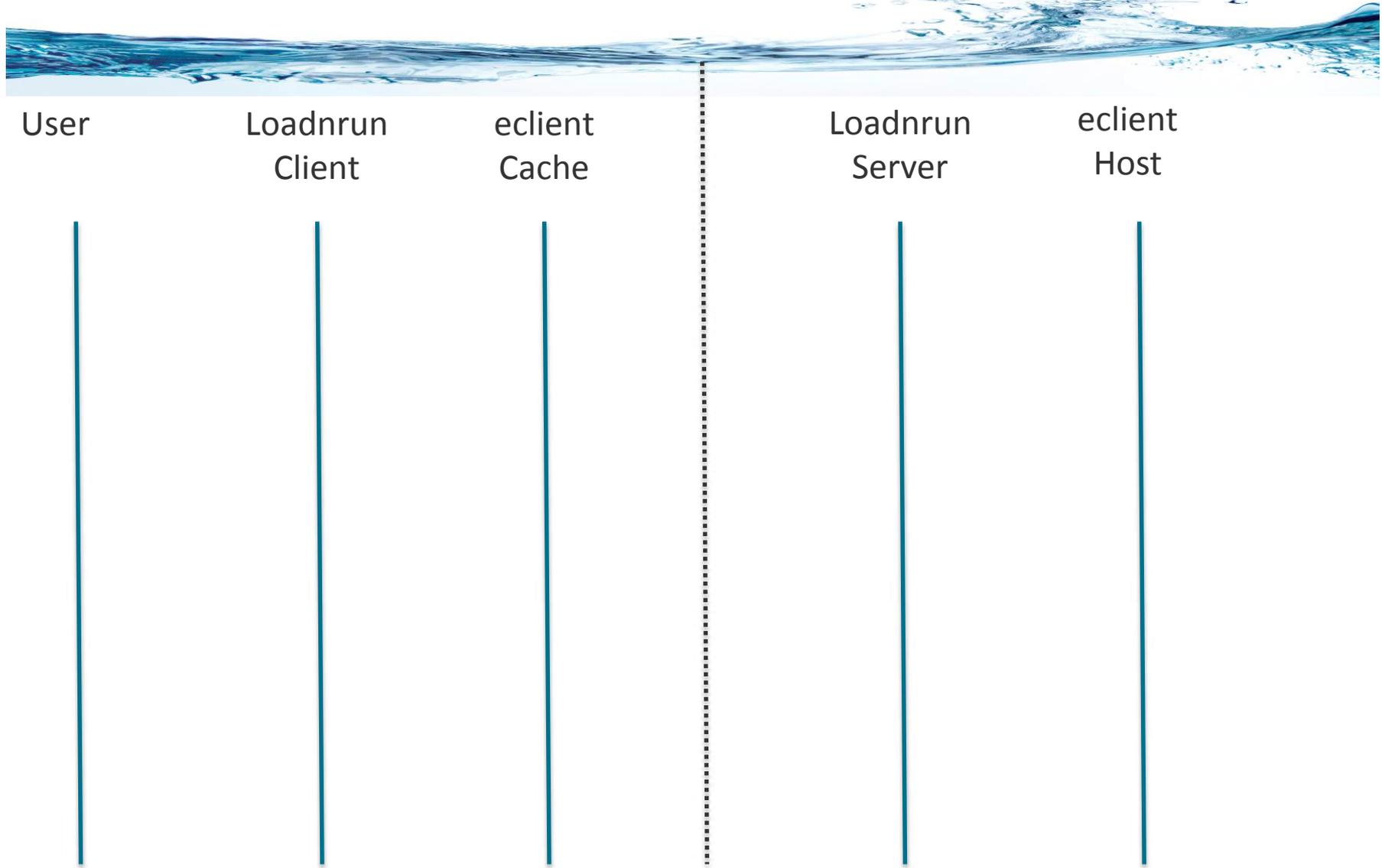
Loadrun Launch – Use Local eclient cache version



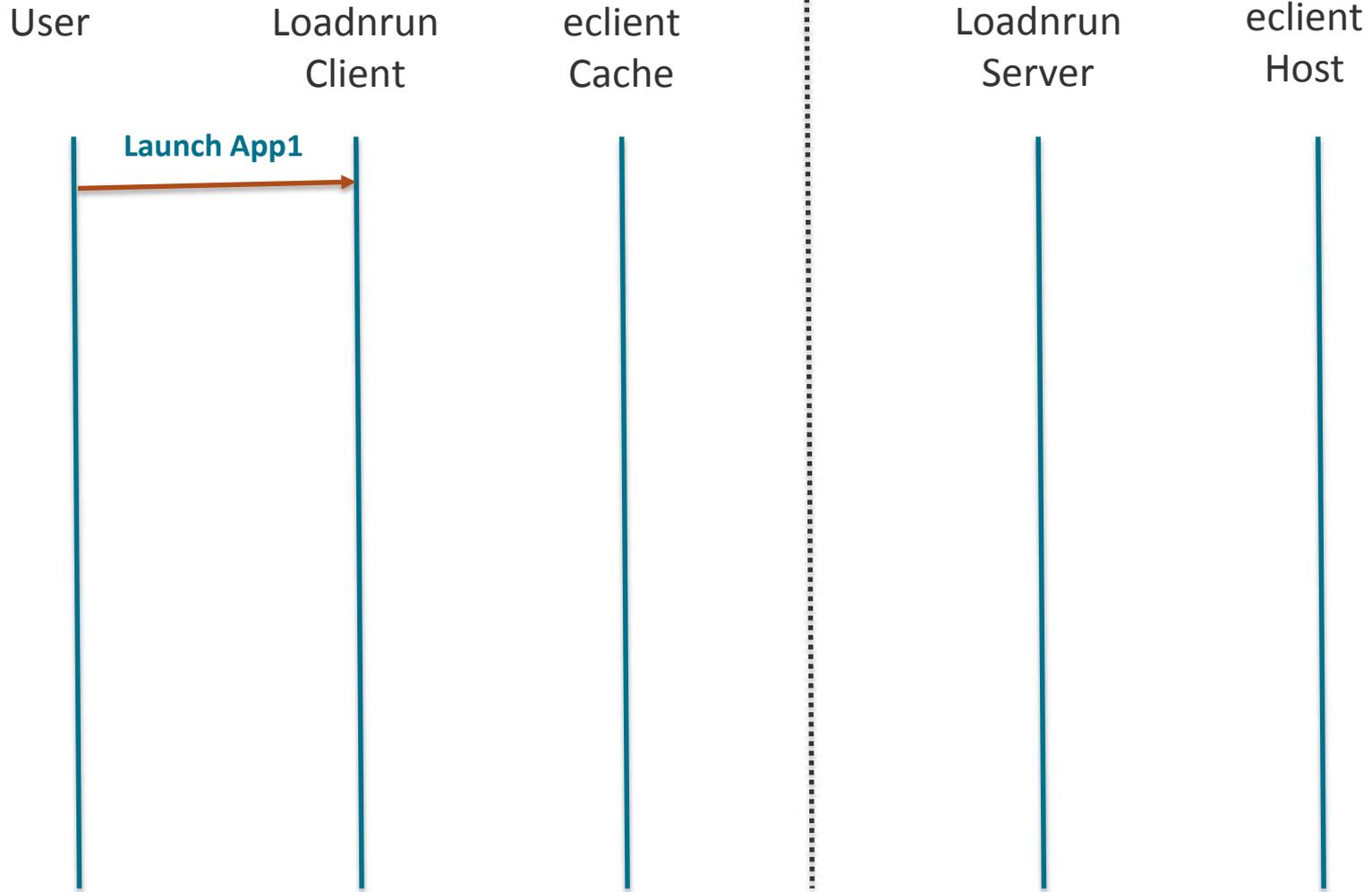
Launching Application: Cache Miss on Client



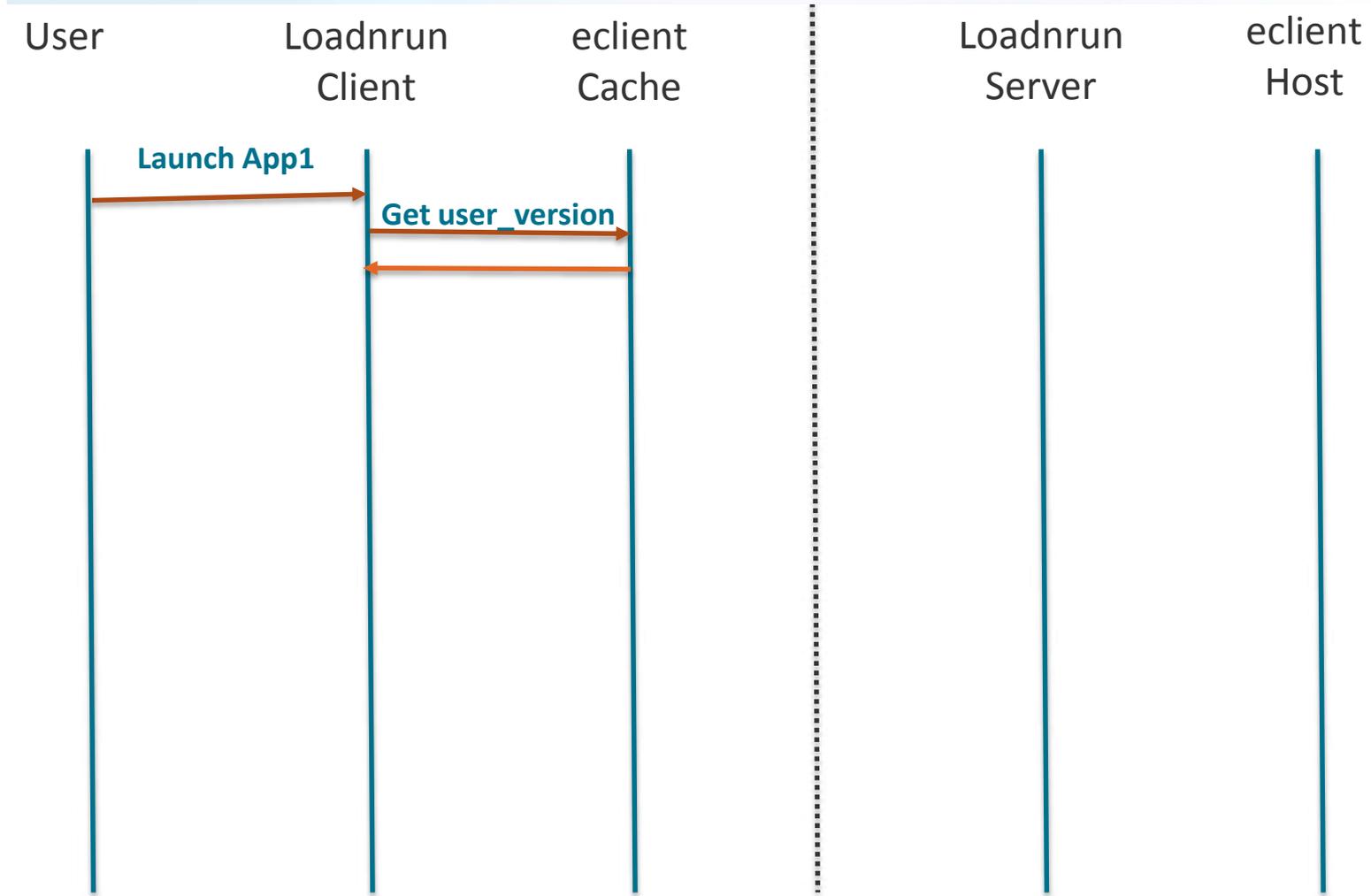
Loadrun Launch – Get eclient Host version



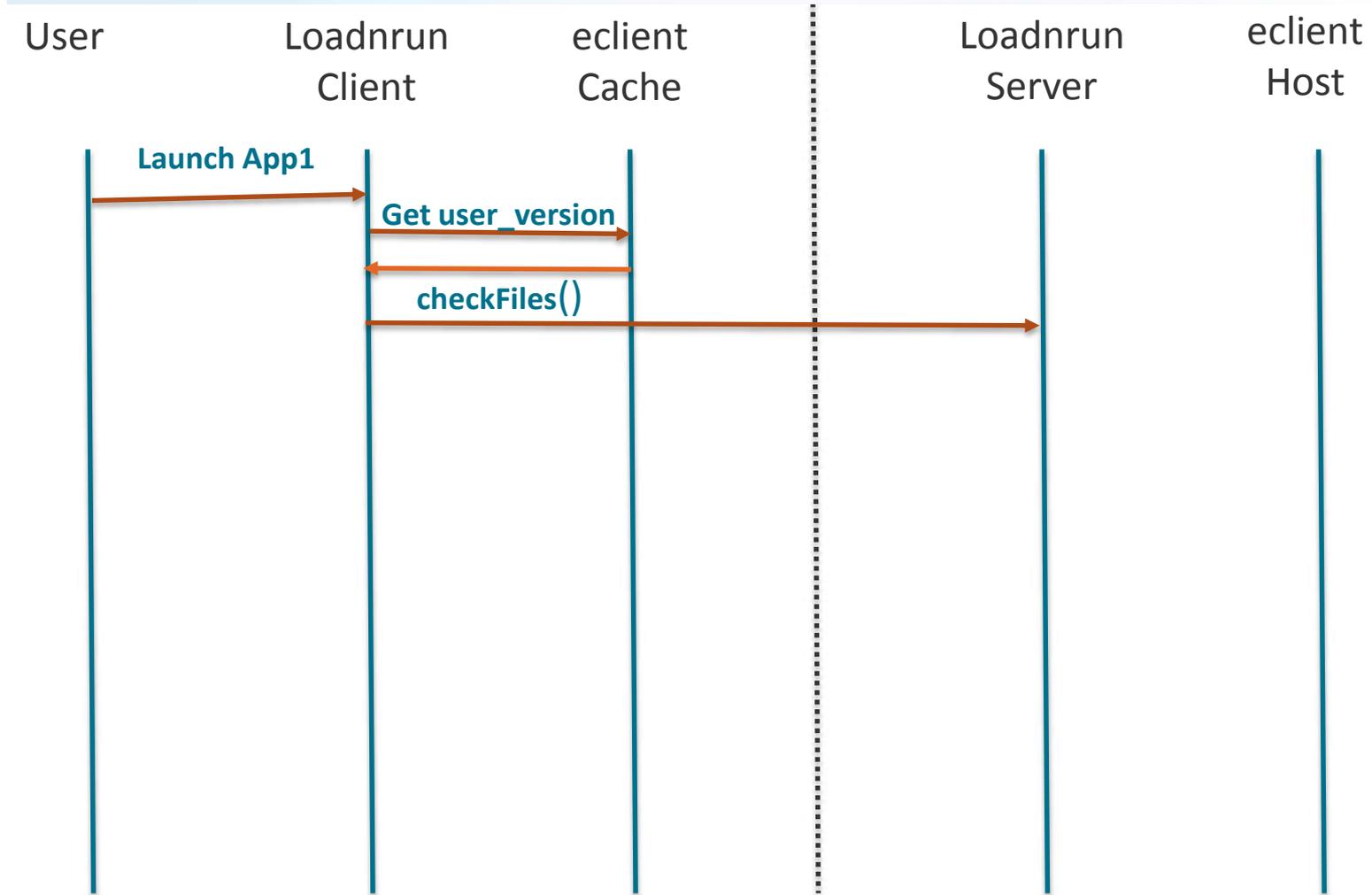
Loadrun Launch – Get eclient Host version



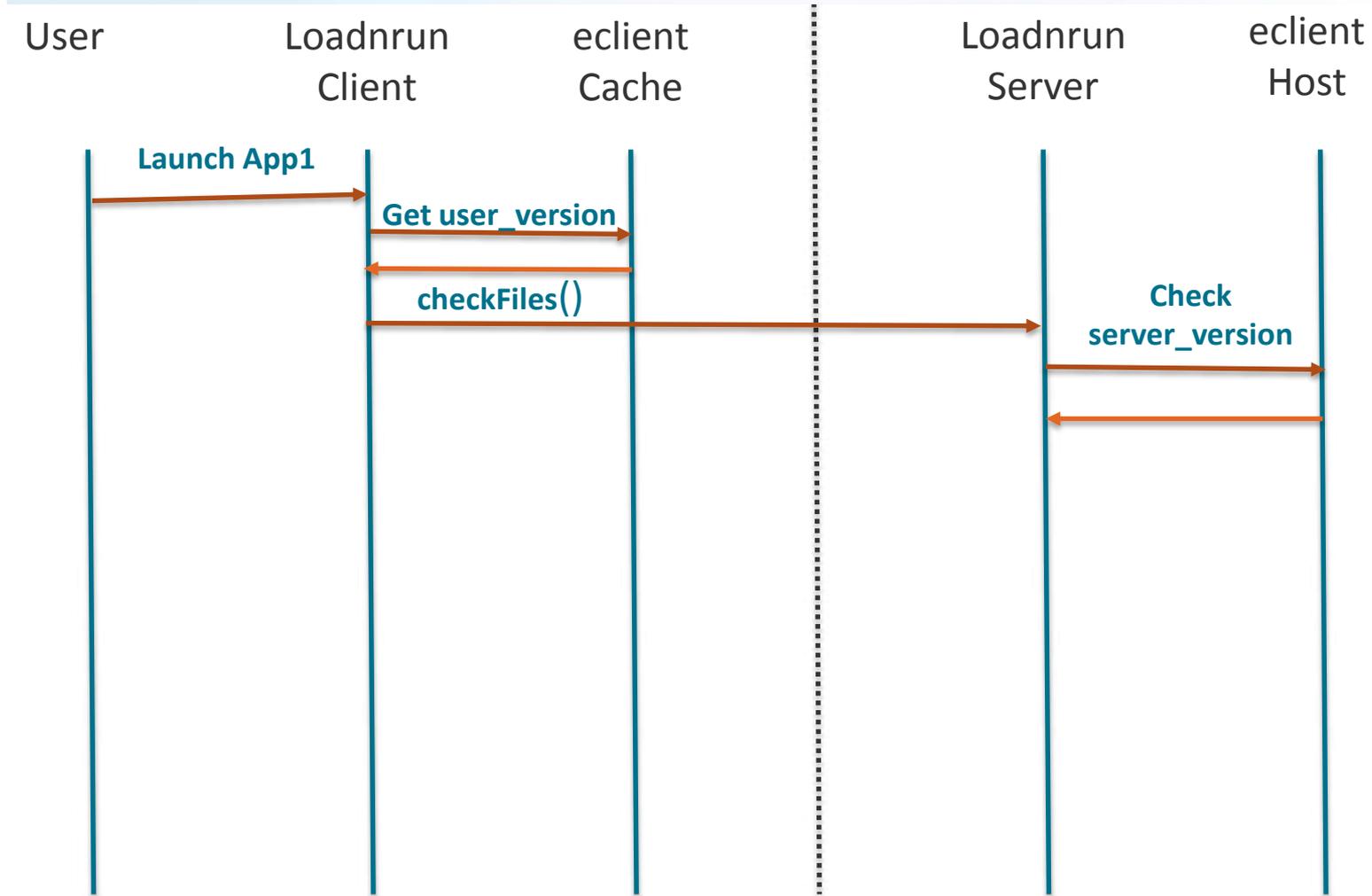
Loadrun Launch – Get eclient Host version



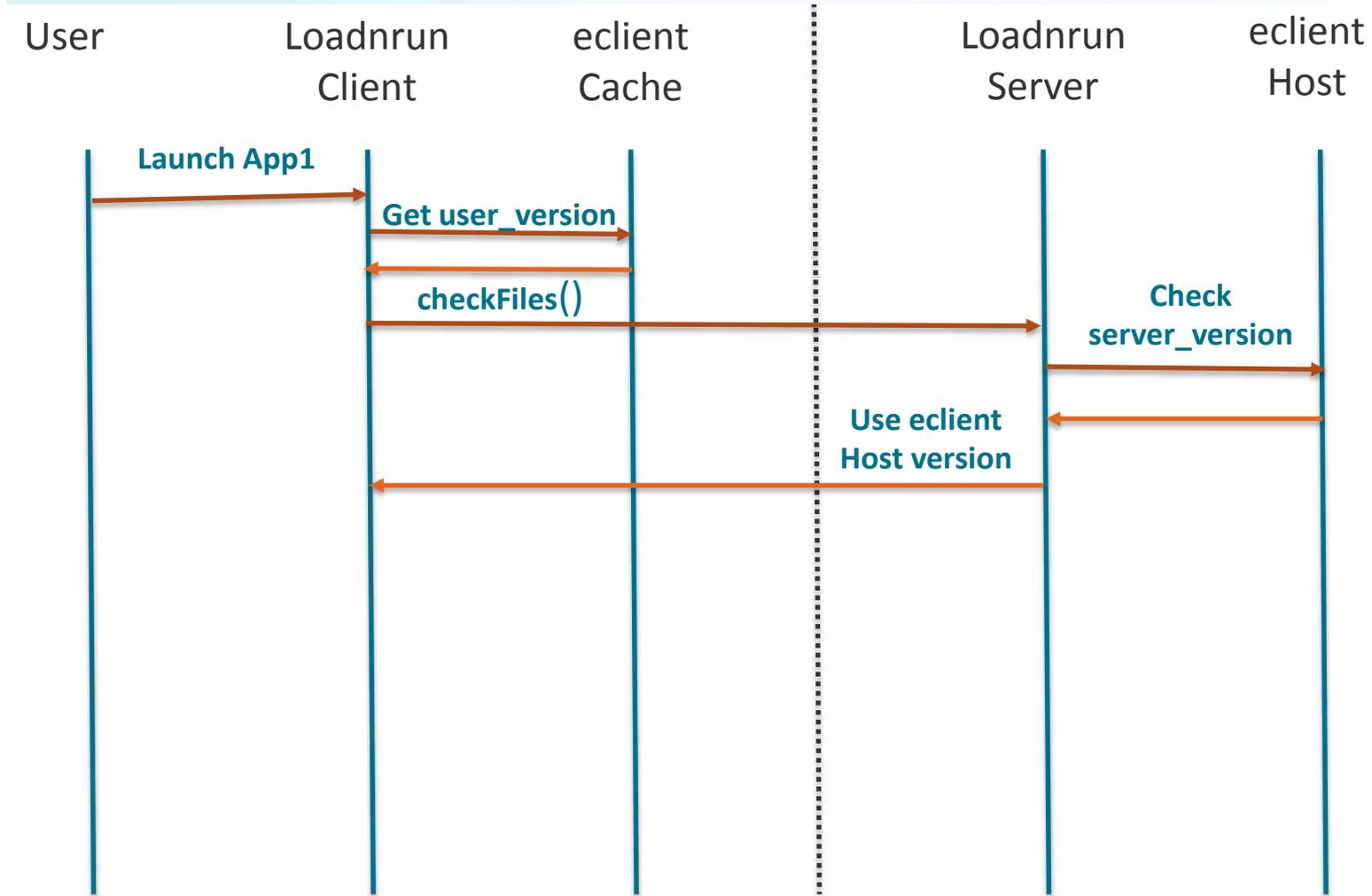
Loadrun Launch – Get eclient Host version



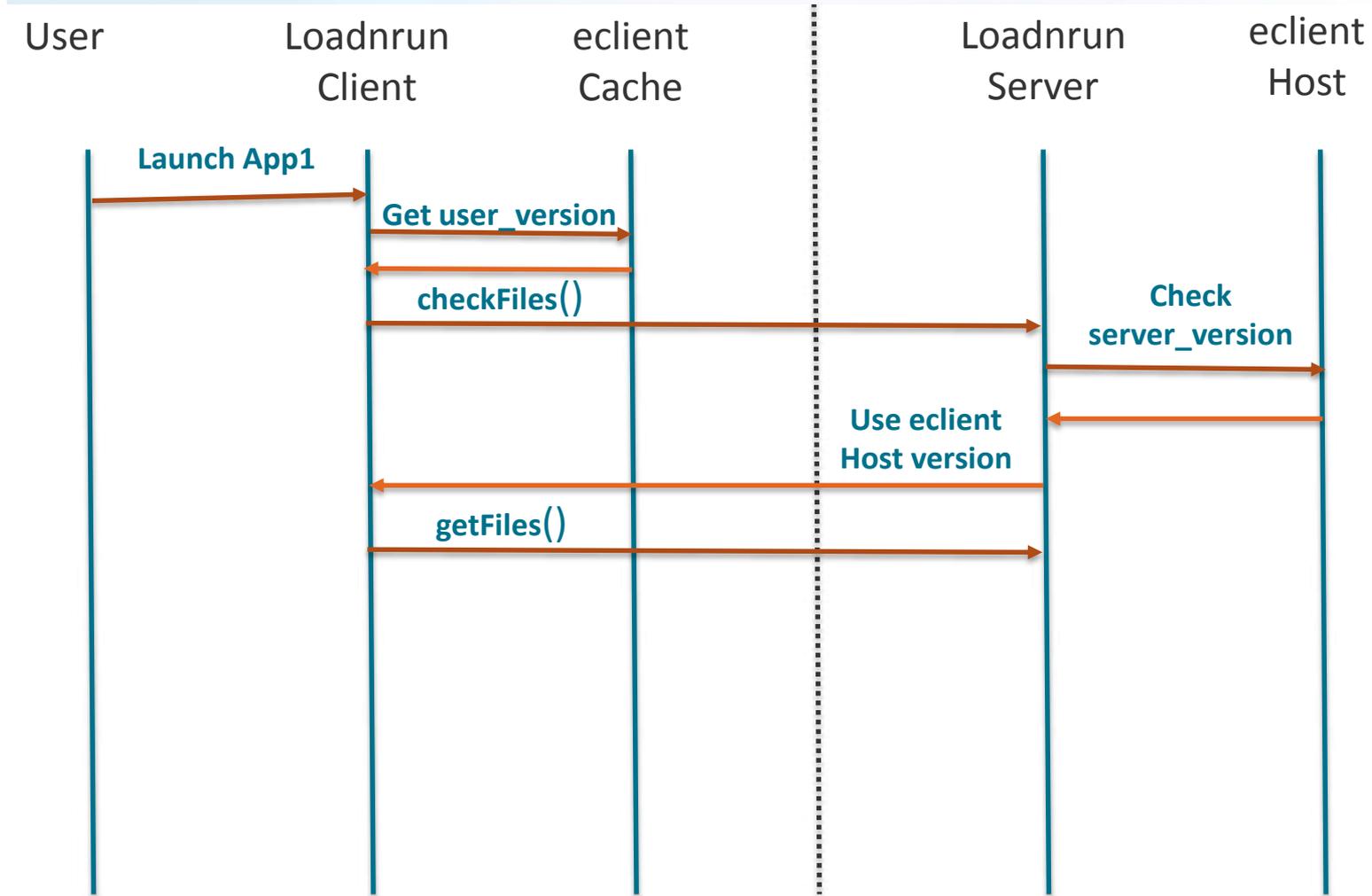
Loadrun Launch – Get eclient Host version



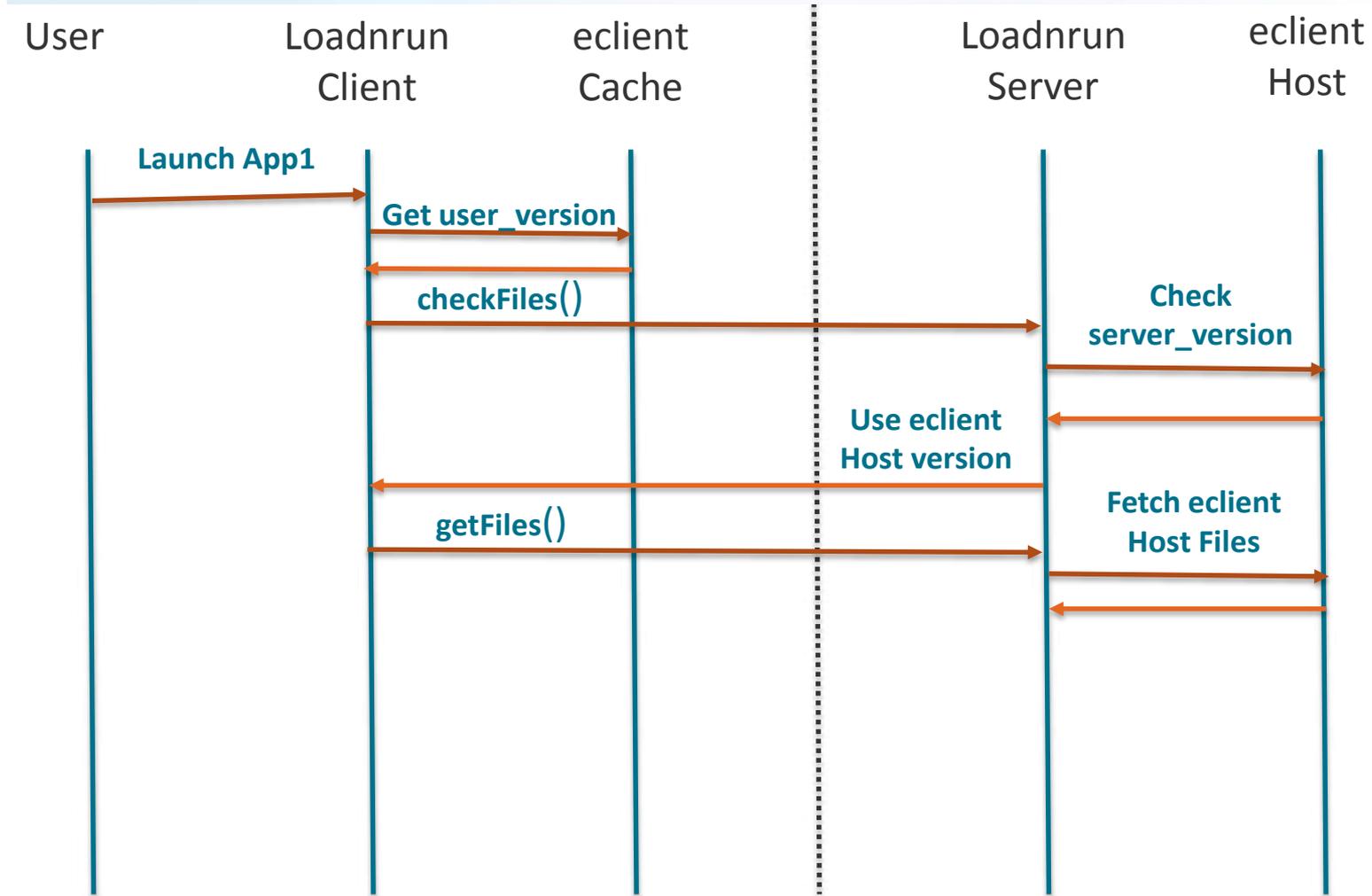
Loadrun Launch – Get eclient Host version



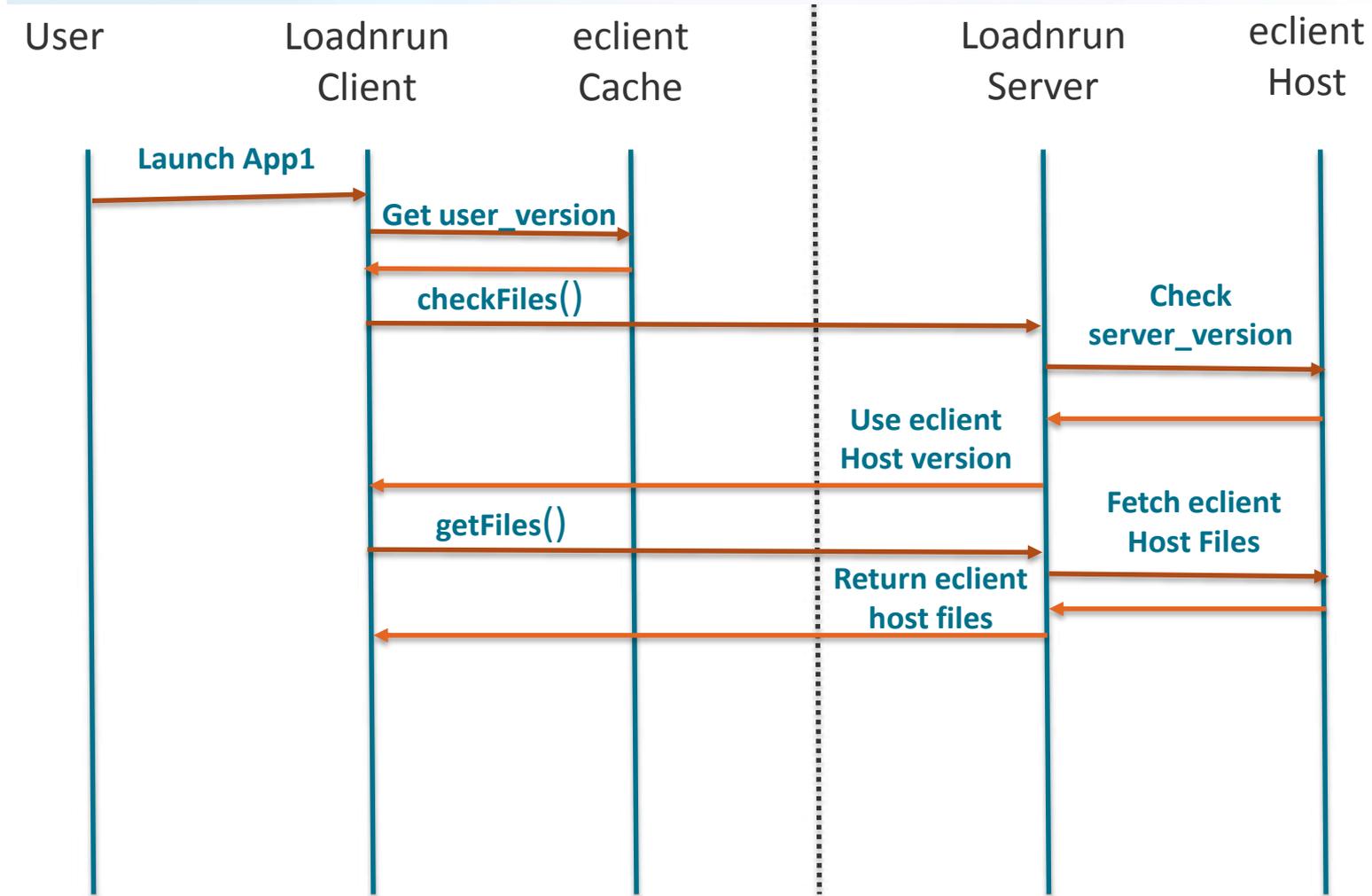
Loadrun Launch – Get eclient Host version



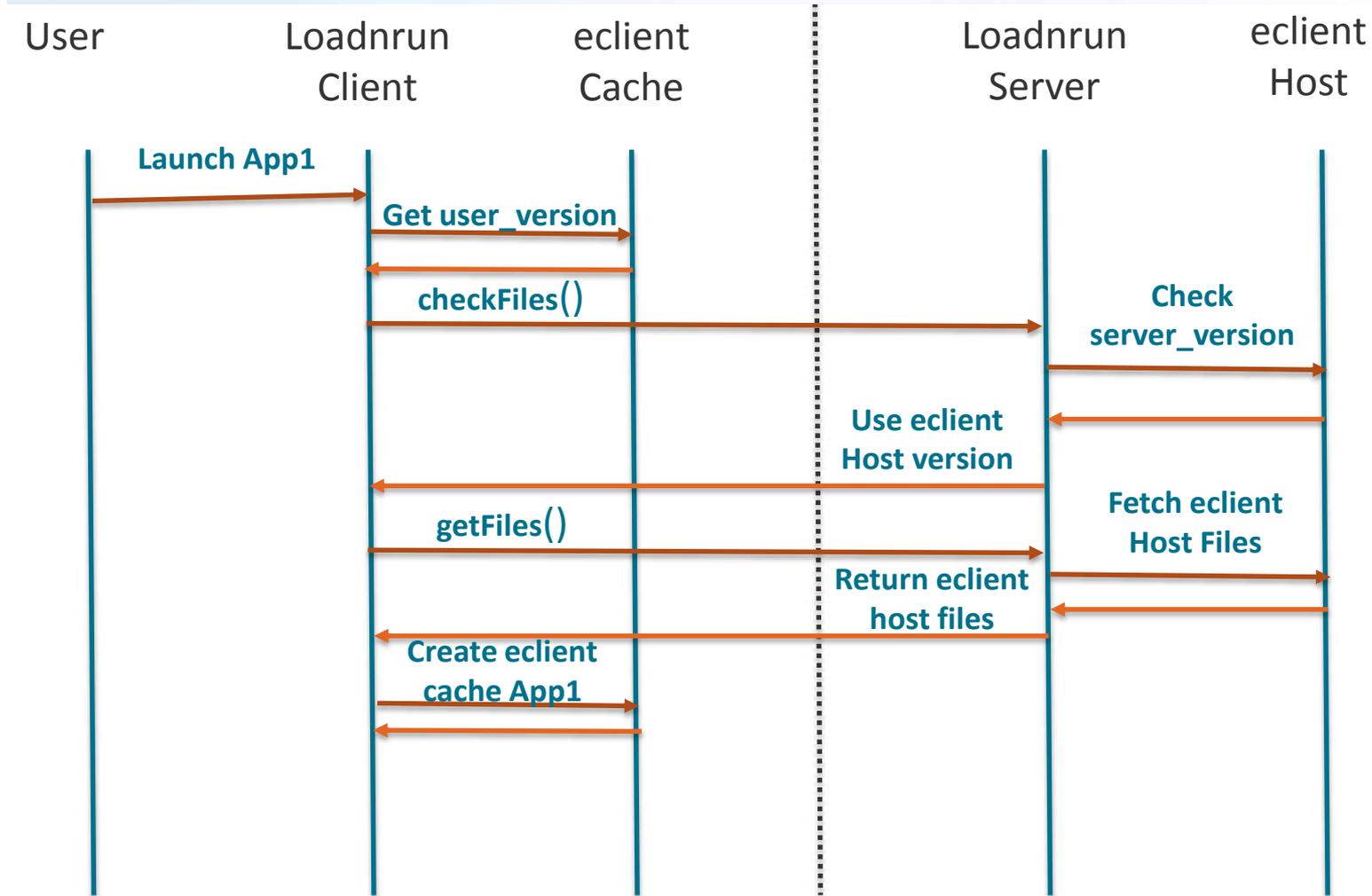
Loadrun Launch – Get eclient Host version



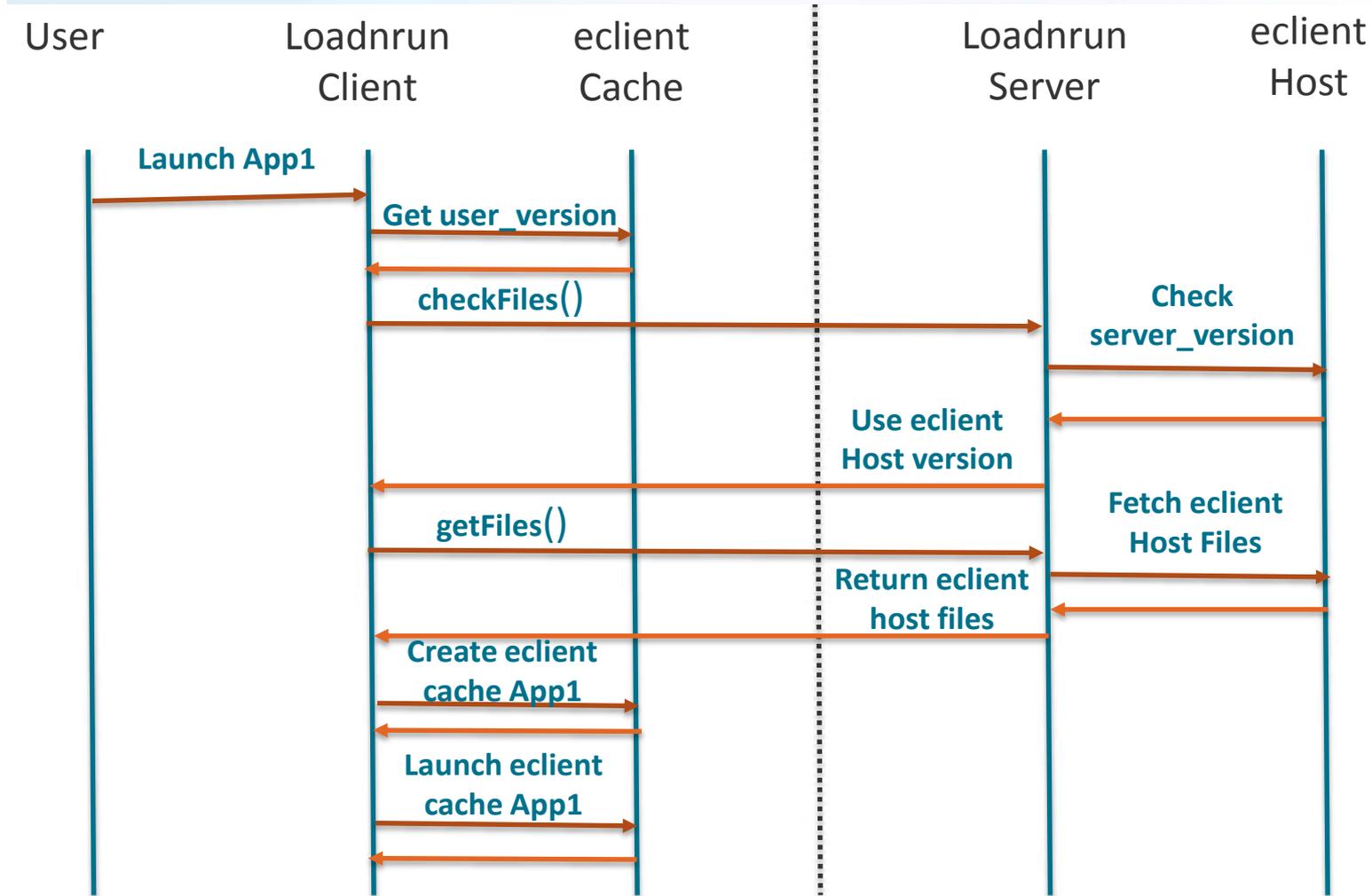
Loadrun Launch – Get eclient Host version



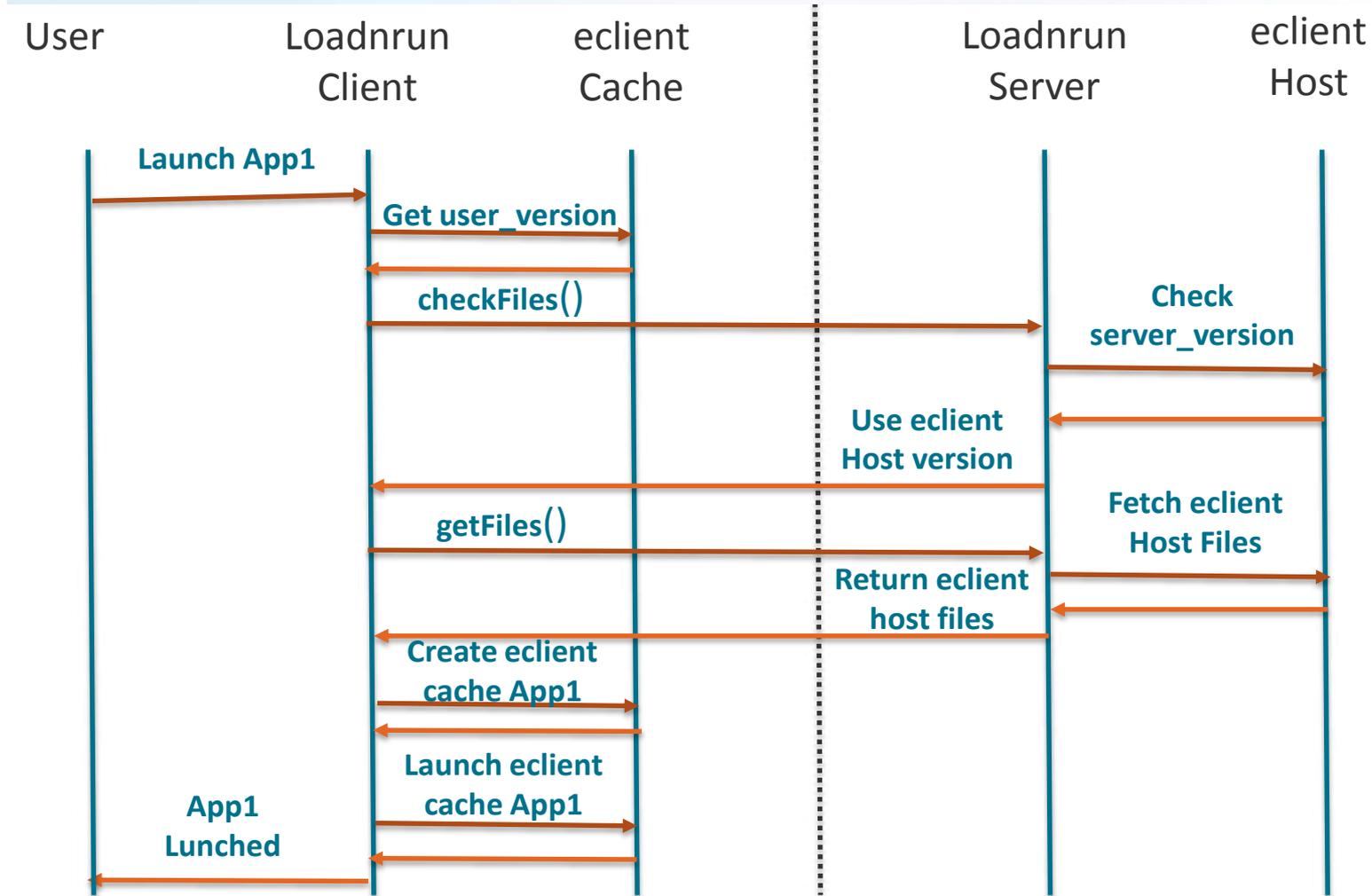
Loadrun Launch – Get eclient Host version



Loadrun Launch – Get eclient Host version



Loadrun Launch – Get eclient Host version



Demo ...



Loadnrun: Multiple Versions



Multiple Versions of Loadrun



- ➔ Each Loadrun runtime can
 - Access the Loadrun Server
 - Download a hosted application if it exists
 - Launch the application using the correct runtime
- ➔ In order for a Loadrun 6.2 runtime to launch a Loadrun 5.1 or Loadrun 6.0
 - The application must exist on the Loadrun Server under the eClient sub-directory
 - The appropriate Loadrun runtime must exist on the Client machine.
- ➔ This also means that OpenROAD 5.1, OpenROAD 6.0 and OpenROAD 6.2 applications and test and production versions can co-exist on the same client machine and be launched and run independent of each other
- ➔ This behavior is helpful in migrations from older OpenROAD versions to newer OpenROAD versions (for example OR 5.1 to OR 6.2)

Loadnrun Server client files



- The Loadnrun Server hosts the files under the following directory
 - %II_LOADNRUN62_W4GLAPPS_DIR%\eClient
- This directory contains sub-directories that correspond to the Loadnrun Suffix value
 - 62demo
 - 62prod
 - 62test
- Note that the value of this suffix does not have anything to do with the version of the OpenROAD applications

%II_LOADNRUN_W4GLAPPS_DIR%\eClient

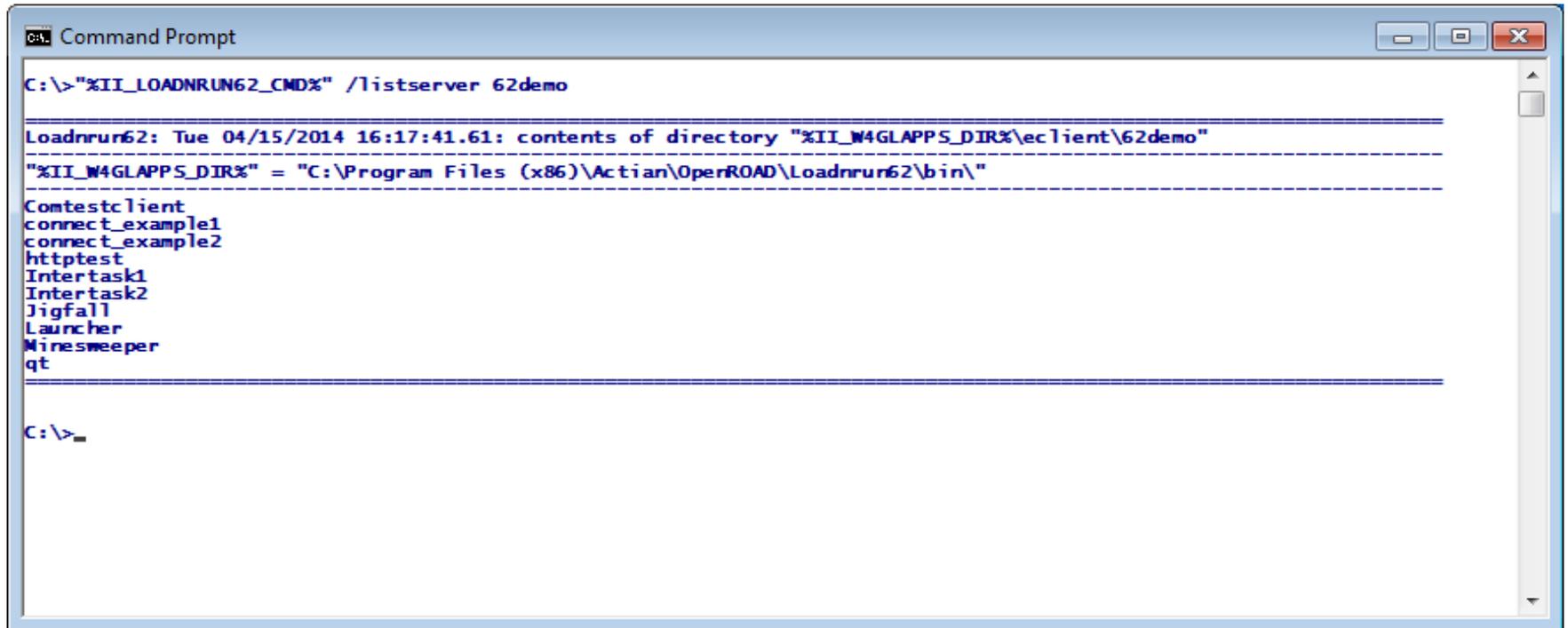


→ The subdirectories in this location is as follows

- 51demo
- 51prod
- 51test
- 60demo
- 60prod
- 60test
- 62demo
- 62prod
- 62test

→ Notice that Loadnrun 6.2 is hosting applications from other versions of OpenROAD

"%II_LOADNRUN62_CMD%" /listserver 62demo

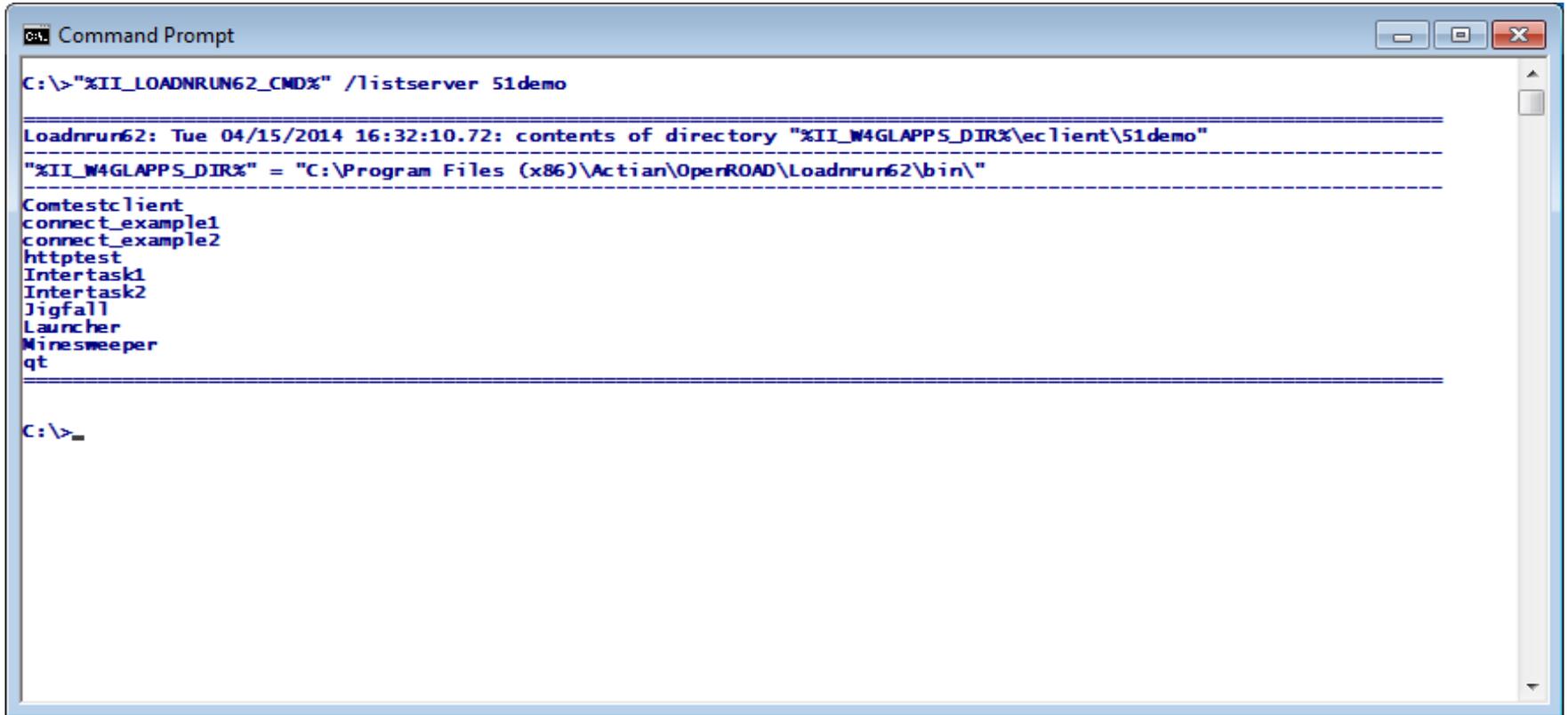


```
Command Prompt
C:\>"%II_LOADNRUN62_CMD%" /listserver 62demo

Loadnrn62: Tue 04/15/2014 16:17:41.61: contents of directory "%II_W4GLAPPS_DIR%\eclient\62demo"
-----
"%II_W4GLAPPS_DIR%" = "C:\Program Files (x86)\Action\OperROAD\Loadnrn62\bin\"
-----
Comtestclient
connect_example1
connect_example2
httpstest
Intertask1
Intertask2
Jigfall
Launcher
Minesweeper
qt
-----

C:\>
```

"%II_LOADNRUN62_CMD%" /listserver 51demo

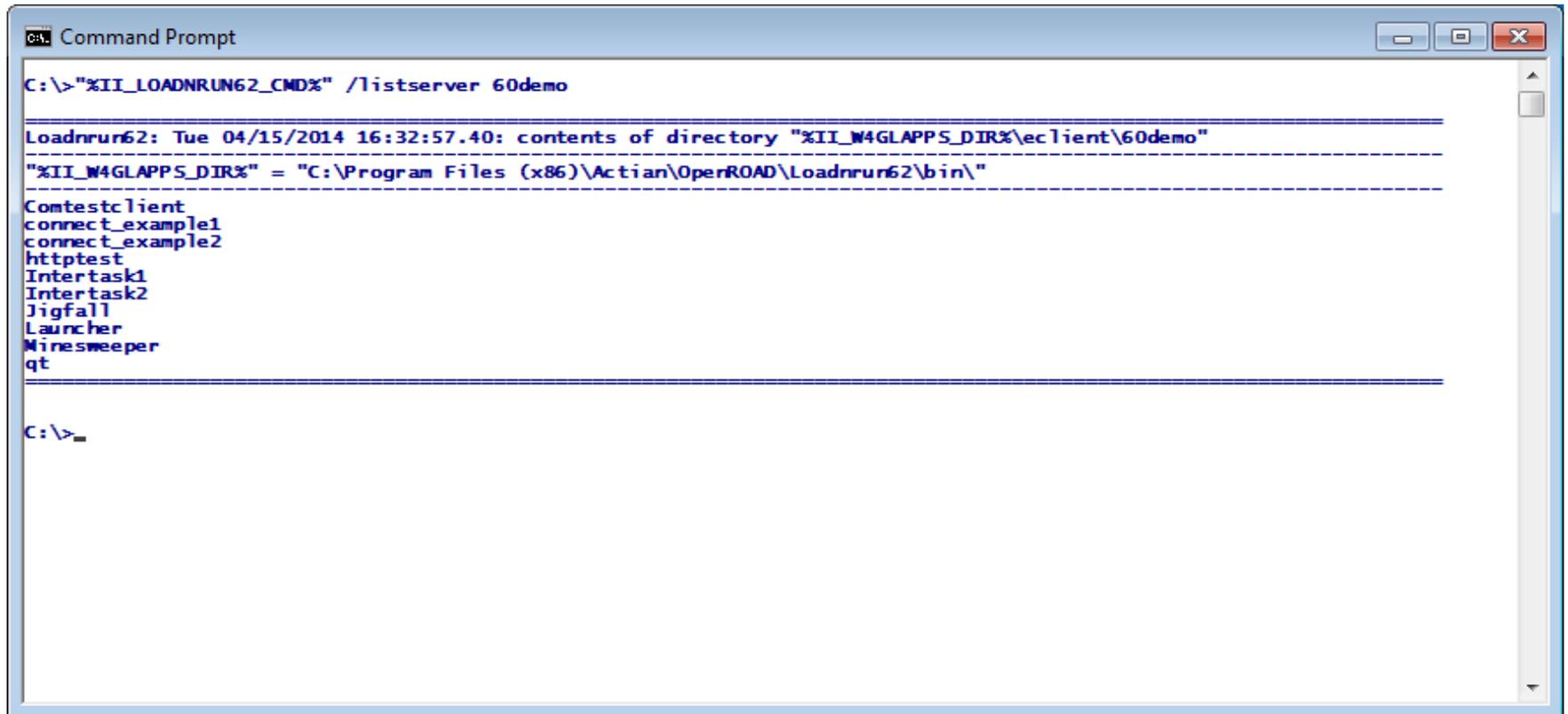


```
C:\>"%II_LOADNRUN62_CMD%" /listserver 51demo

Loadnr62: Tue 04/15/2014 16:32:10.72: contents of directory "%II_W4GLAPPS_DIR%\eclient\51demo"
-----
"%II_W4GLAPPS_DIR%" = "C:\Program Files (x86)\Actian\OperROAD\Loadnr62\bin\"
-----
Comtestclient
connect_example1
connect_example2
httpstest
Intertask1
Intertask2
Jigfall
Launcher
Minesweeper
qt
-----

C:\>
```

"%II_LOADNRUN62_CMD%" /listserver 60demo

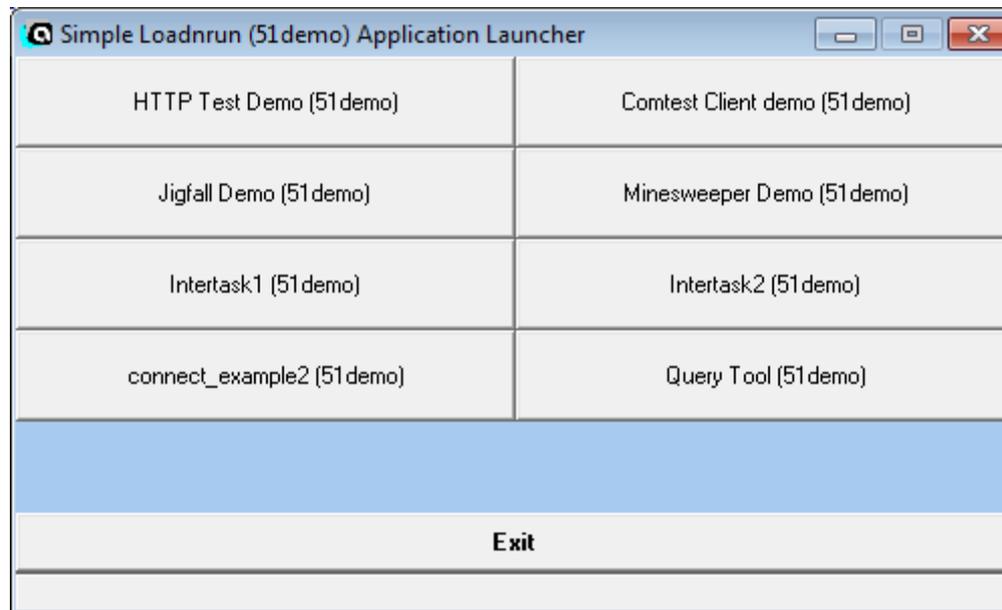


```
C:\>"%II_LOADNRUN62_CMD%" /listserver 60demo

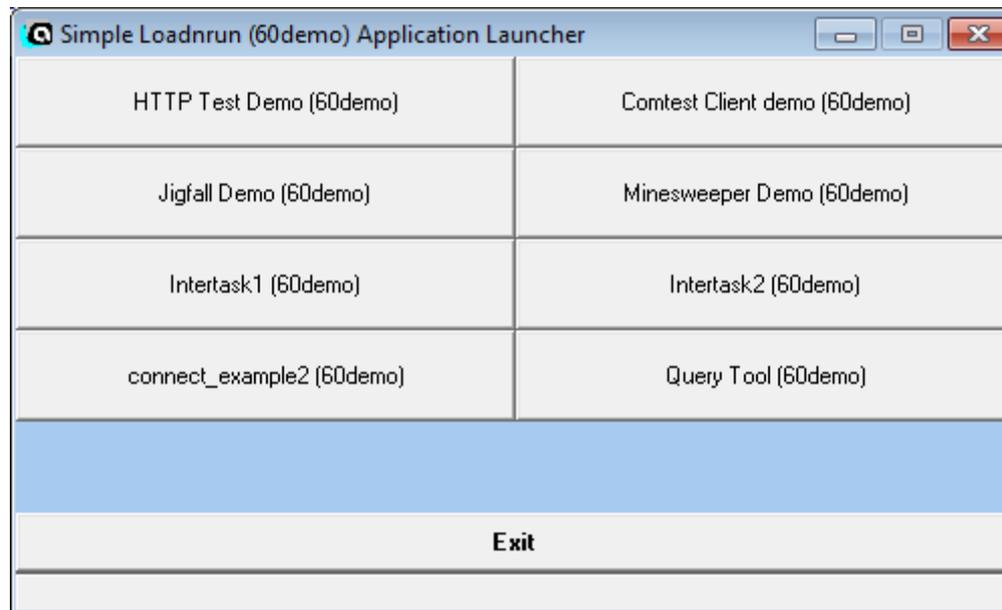
Loadnr62: Tue 04/15/2014 16:32:57.40: contents of directory "%II_W4GLAPPS_DIR%\eclient\60demo"
-----
"%II_W4GLAPPS_DIR%" = "C:\Program Files (x86)\Actian\OperROAD\Loadnr62\bin\"
-----
Comtestclient
connect_example1
connect_example2
httpstest
Intertask1
Intertask2
Jigfall
Launcher
Minesweeper
qt
-----

C:\>
```

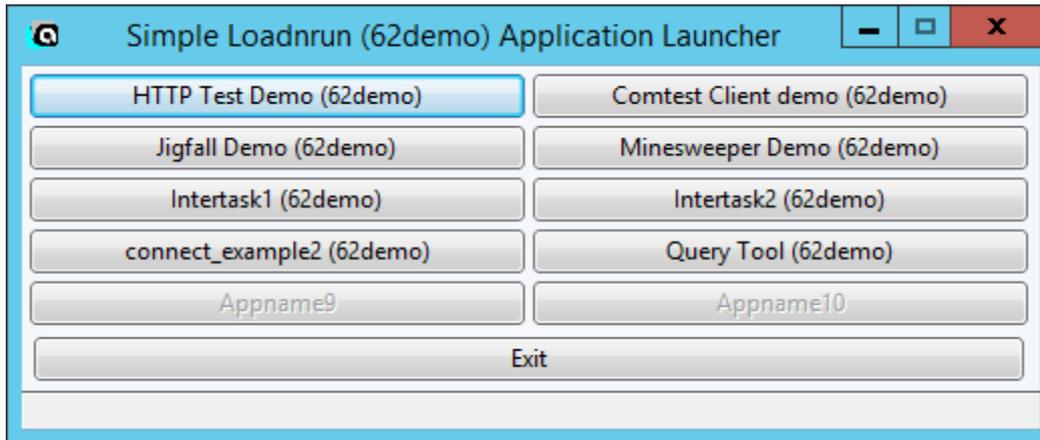
"%II_LOADNRUN51_CMD%" launcher %II_LOADNRUN51_GATEWAY% 51demo



"%II_LOADNRUN51_CMD%" launcher %II_LOADNRUN51_GATEWAY% 60demo



"%II_LOADNRUN51_CMD%" launcher %II_LOADNRUN51_GATEWAY% 62demo



Configure the IIS Gatekeeper



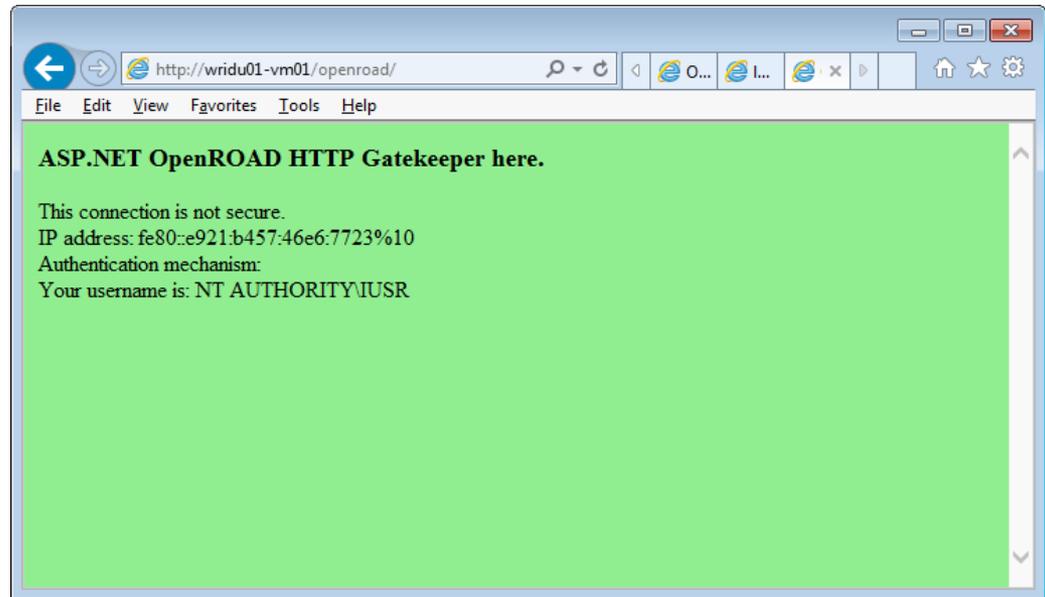
OpenROAD Gatekeeper



- The OpenROAD Gatekeeper needs to be configured for Loadrun.
 - This provide HTTP/HTTPS access to the OpenROAD Server
 - Uses a Web Server application
- There are three versions of the OpenROAD Gatekeeper
 - IIS ASP.NET version (IIS 6 or IIS 7.x)
 - Tomcat 32-bit Java version
 - Tomcat 64-bit Java version
- In this section the IIS 7.0 ASP.NET version is used
- All versions have been configured using the existing OpenROAD procedures and documentation

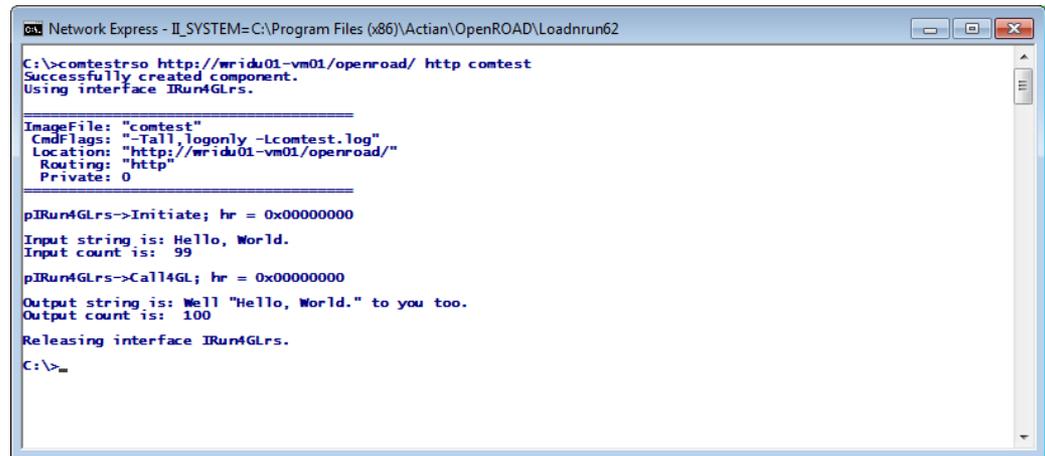
The application can be launched via the URL

- When the URL for the gatekeeper is passed, then it is launched
- The machine name is passed explicitly rather than localhost



Test access to OpenROAD Server via Gatekeeper

- The comtestrso command can be used to verify access to the OpenROAD Server via the Gatekeeper
- This test verifies that requests can be made to the OpenROAD Server via an HTTP request



```
Network Express - II_SYSTEM=C:\Program Files (x86)\Action\OpenROAD\Loadrun62
C:\>comtestrso http://wridu01-vm01/openroad/ http comtest
Successfully created component.
Using interface IRun4GLrs.

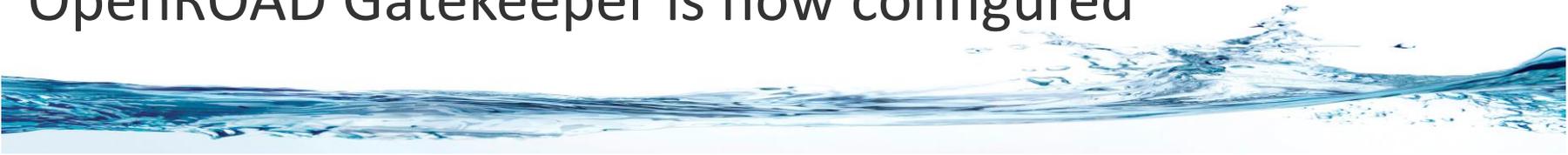
-----
ImageFile: "comtest"
CmdFlags: "-Tall,logonly -Lcomtest.log"
Location: "http://wridu01-vm01/openroad/"
Routing: "http"
Private: 0
-----

pIRun4GLrs->Initiate; hr = 0x00000000
Input string is: Hello, World.
Input count is: 99

pIRun4GLrs->Call4GL; hr = 0x00000000
Output string is: We'll "Hello, World." to you too.
Output count is: 100

Releasing interface IRun4GLrs.
C:\>_
```

OpenROAD Gatekeeper is now configured



- The platforms that have been tested are
 - Windows 7 (32-bit/64-bit)
 - Windows 8 (32-bit/64-bit), Windows 8.1 (32-bit/64-bit)
 - Windows Server 2008 R2 (64-bit)
 - Windows Server 2012 (64-bit)

- OpenROAD and Loadrun Versions tested are
 - OpenROAD 5.1.1+, Loadrun 5.1
 - OpenROAD 6.0.2+, Loadrun 6.0
 - OpenROAD 6.2.0, Loadrun 6.2

Configure the Tomcat 64-bit Gatekeeper



OpenROAD Gatekeeper



- The OpenROAD Gatekeeper needs to be configured for Loadrun.
 - This provide HTTP/HTTPS access to the OpenROAD Server
 - Uses a Web Server application
- There are three versions of the OpenROAD Gatekeeper
 - IIS ASP.NET version (IIS 6 or IIS 7.x)
 - Tomcat 32-bit Java version
 - Tomcat 64-bit Java version
- In this section the Tomcat 64-bit Java version is used
- All versions have been configured using the existing OpenROAD procedures and documentation

The %II_SYSTEM%\ingres\bin64 directory (1 or 2)



- Can be used to access the OpenROAD server (32-bit) from 64-bit processes (e.g. a JVM) on Windows 64-bit installations.
- The files contained in this directory are
 - ororso.dll
 - orps.dll
 - orrsojni.dll
 - comtestrso.exe
- These are 64-bit versions of those delivered in %II_SYSTEM%\ingres\bin.

The %II_SYSTEM%\ingres\bin64 directory (2 or 2)



- The Microsoft Visual C++ runtime redistributables (64-bit) are
 - mfc100.dll
 - mfc100u.dll
 - mfcm100.dll
 - mfcm100u.dll
 - msvcpl100.dll
 - msvcr100.dll
- This runtime is needed to support the 64-bit OpenROAD Deliverables.

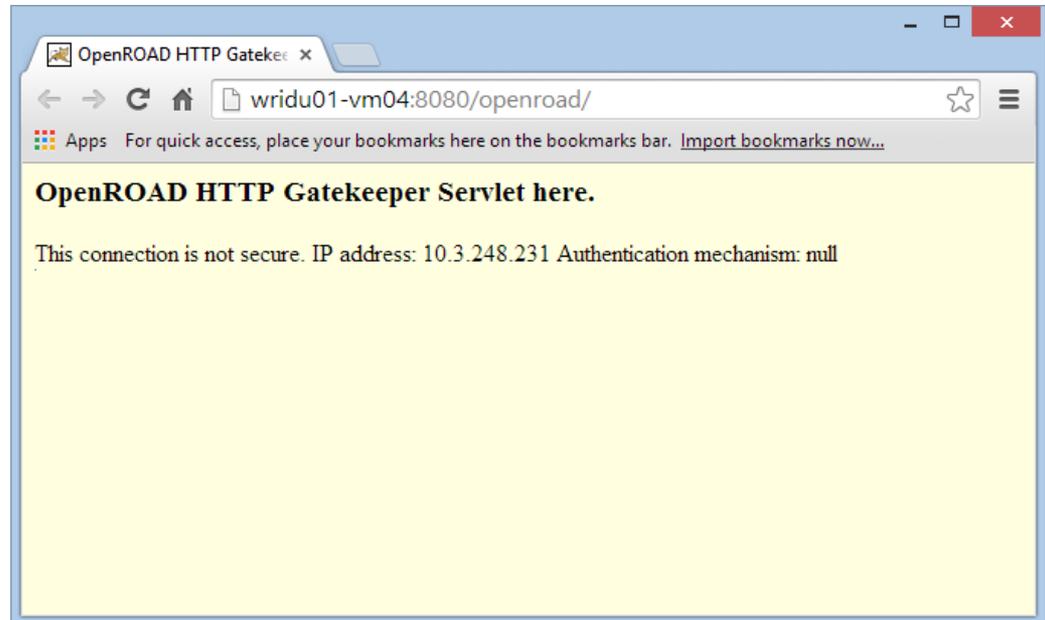
Configuring the 64-bit OpenROAD Deliverables for use with Tomcat



- Use the files contained in the %II_SYSTEM%\ingres\bin64 directory
- Register the bin64-DLLs orrso.dll and orps.dll using %windir%\system32\regsvr32 (64-bit version)
 - %windir%\system32\regsvr32 %II_SYSTEM%\ingres\bin64\orps.dll
 - %windir%\system32\regsvr32 %II_SYSTEM%\ingres\bin64\orrso.dll
- Add %II_SYSTEM%\ingres\bin64 directory in front of the PATH (or other) environment variable(s) where required (when using 64-bit processes)
 - For Tomcat configure its “Java” settings – add the Java option:
 - -Djava.library.path=<full path of %II_SYSTEM%\ingres\bin64>

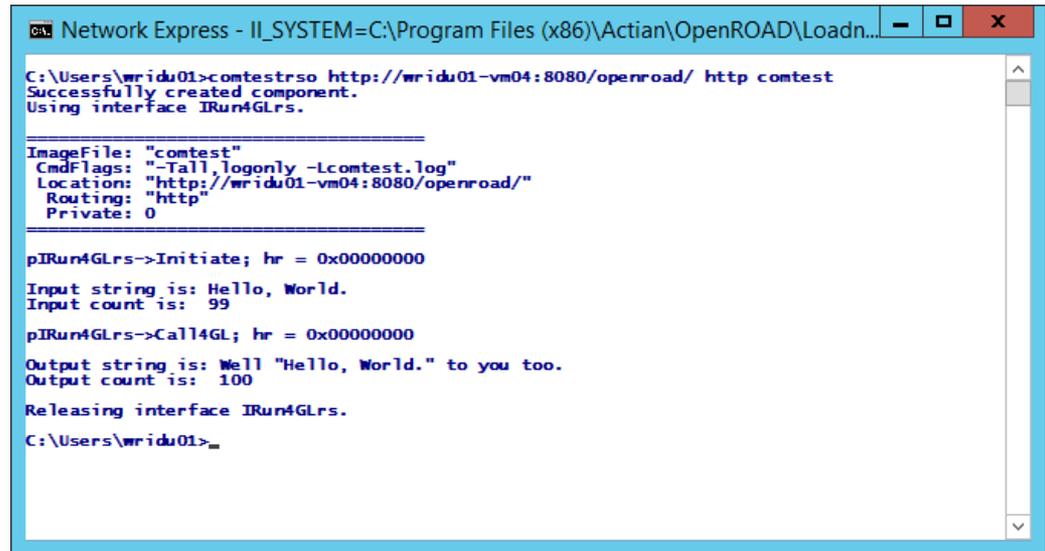
The application can be launched via the URL

- When the URL for the gatekeeper is passed, then it is launched
- The machine name is passed explicitly rather than localhost



Test access to OpenROAD Server via Gatekeeper

- The comtestrso command can be used to verify access to the OpenROAD Server via the Gatekeeper
- This test verifies that requests can be made to the OpenROAD Server via an HTTP request



```
C:\Users\wridu01>comtestrso http://wridu01-vm04:8080/openroad/ http comtest
Successfully created component.
Using interface IRun4GLrs.

=====
ImageFile: "comtest"
CmdFlags: "-Tall,logonly -Lcomtest.log"
Location: "http://wridu01-vm04:8080/openroad/"
Routing: "http"
Private: 0
=====

pIRun4GLrs->Initiate; hr = 0x00000000
Input string is: Hello, World.
Input count is: 99

pIRun4GLrs->Call4GL; hr = 0x00000000
Output string is: Well "Hello, World." to you too.
Output count is: 100

Releasing interface IRun4GLrs.
C:\Users\wridu01>
```

Install and Configure Loadrun



Install Loadrun

- Launch the Loadrun 6.2 installer
- Enter information as shown in the frame to the right
- This options will configure the Loadrun Server and the Loadrun Client on the same machine
- The Server Suffix is used to specify the default location of user applications

Action Pre-Installer Wizard

Input Loadrun Configuration Parameters
Input the parameters that will be used to Configure Loadrun

Customization

Appname: laucher
Gateway URL: http://wridu01-vm01/openroad/
Server Suffix: 62demd

Client Options

eClient Runtime: yes
Launcher Shortcut: no
Default Appname Shortcut:
Install S/A Net Client: yes
Demo Shortcuts: yes

Server Options

Sample Demos: yes
Demo Source: yes

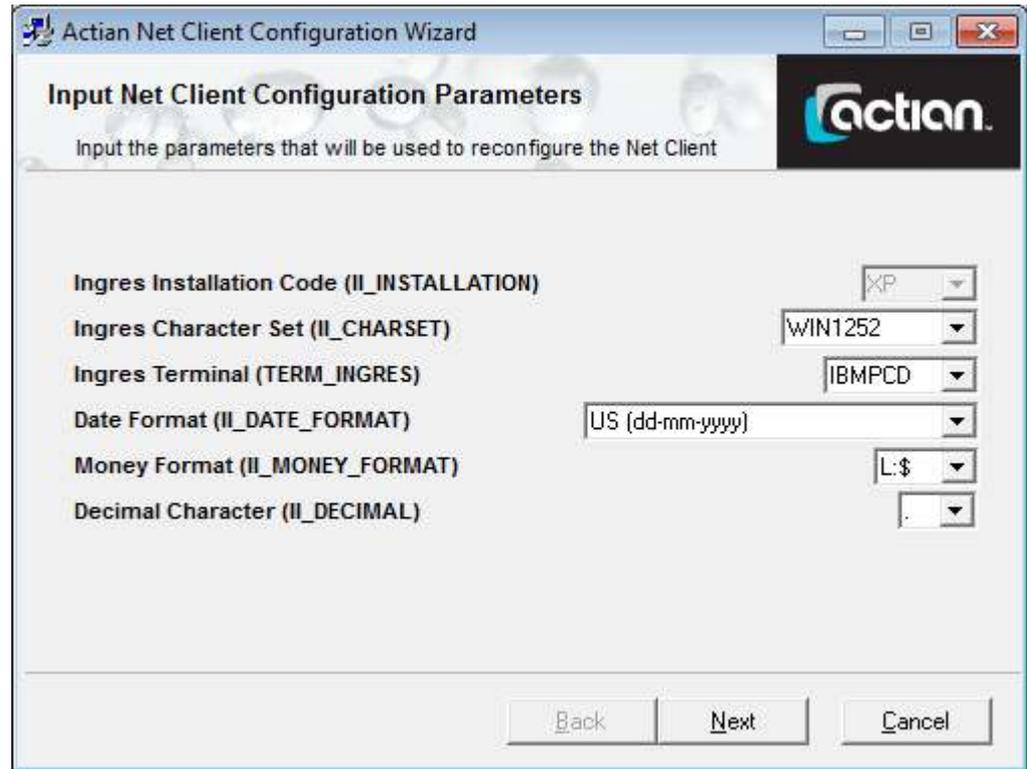
Common Options

Product Uninstall Shortcut: yes

Back Next Cancel

Accept of modify S/A Net Options

- The Ingres 10.1.1 client has been selected for installation
- The Installation Code cannot be modified but the other options can be modified



The screenshot shows the 'Action Net Client Configuration Wizard' dialog box. The title bar reads 'Action Net Client Configuration Wizard'. The main heading is 'Input Net Client Configuration Parameters' with a sub-instruction: 'Input the parameters that will be used to reconfigure the Net Client'. The 'Action' logo is in the top right corner. The dialog contains several configuration options, each with a dropdown menu:

- Ingres Installation Code (II_INSTALLATION): XP
- Ingres Character Set (II_CHARSET): WIN1252
- Ingres Terminal (TERM_INGRES): IBMPCD
- Date Format (II_DATE_FORMAT): US (dd-mm-yyyy)
- Money Format (II_MONEY_FORMAT): L:\$
- Decimal Character (II_DECIMAL): .

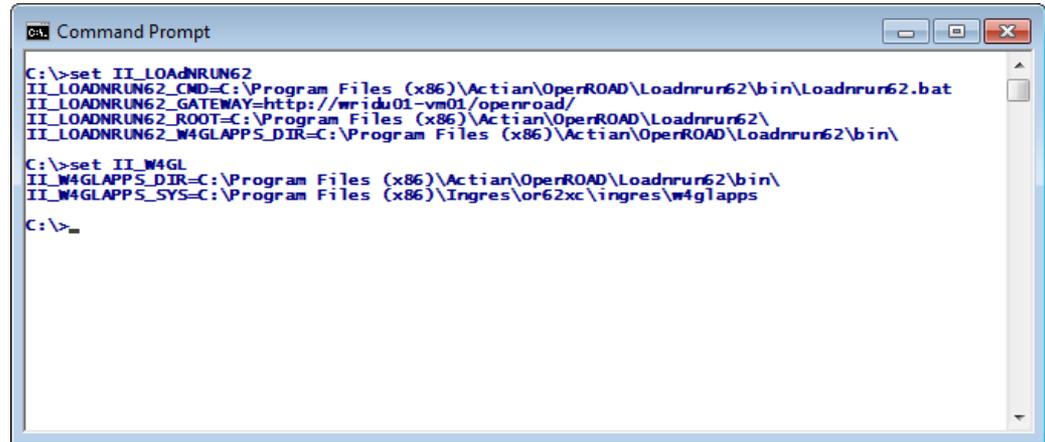
At the bottom of the dialog are three buttons: 'Back', 'Next', and 'Cancel'.

Register Loadnrun to the OpenROAD Server



Verify Environment Variables for Loadrun

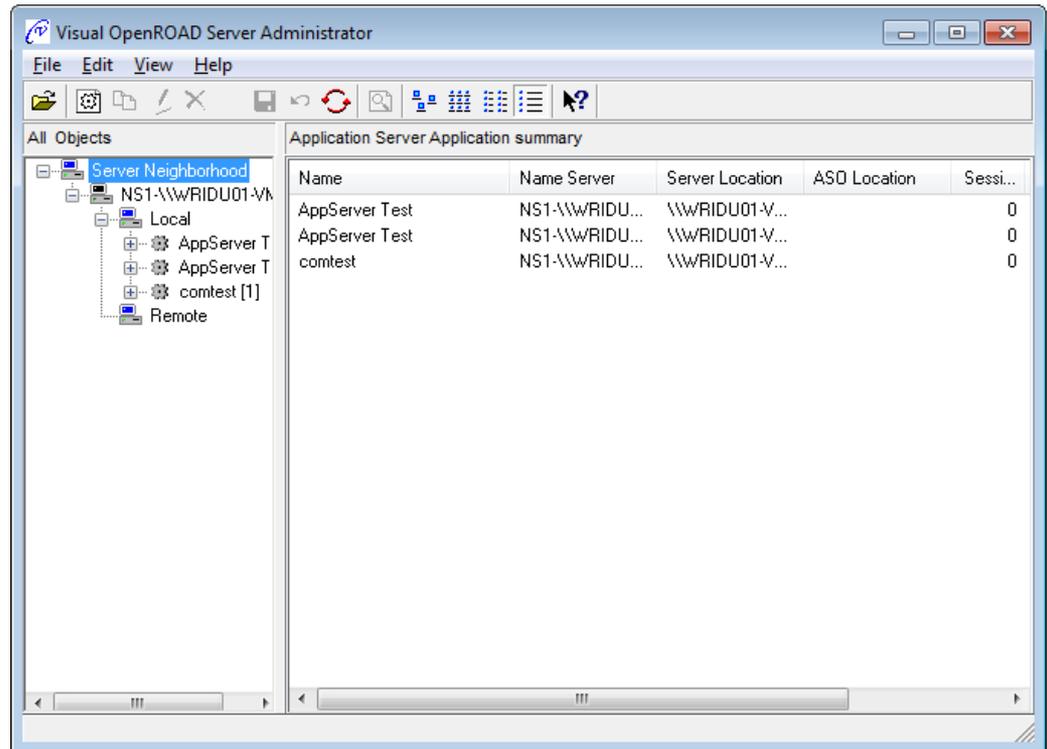
- Verify that the II_LOADNRUN62 environment variables are set
- Verify that the II_W4GLAPPS_SYS is correct
- Verify that the II_W4GLAPPS_DIR is correct and points to the same location as II_LOADNRUN62_W4GLAPPS_DIR
- (The system may need to be rebooted if II_W4GLAPPS_DIR was changed from a previous value in order for the OpenROAD Server to pick it up.)



```
Command Prompt
C:\>set II_LOADNRUN62
II_LOADNRUN62_CMD=C:\Program Files (x86)\Actian\OpenROAD\Loadnr62\bin\Loadnr62.bat
II_LOADNRUN62_GATEWAY=http://wridu01-vm01/openroad/
II_LOADNRUN62_ROOT=C:\Program Files (x86)\Actian\OpenROAD\Loadnr62\
II_LOADNRUN62_W4GLAPPS_DIR=C:\Program Files (x86)\Actian\OpenROAD\Loadnr62\bin\
C:\>set II_W4GL
II_W4GLAPPS_DIR=C:\Program Files (x86)\Actian\OpenROAD\Loadnr62\bin\
II_W4GLAPPS_SYS=C:\Program Files (x86)\Ingres\or62xc\ingres\w4glapps
C:\>
```

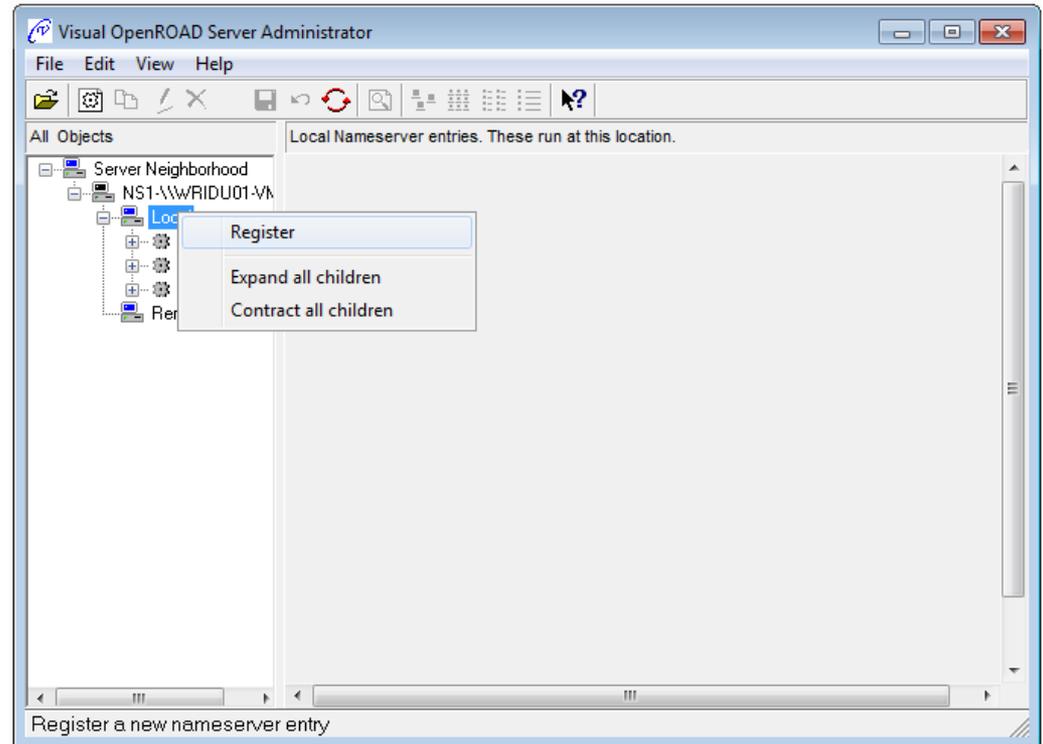
Use VOSA to configure Loadnrun

- ➔ Launch VOSA to configure Loadnrun



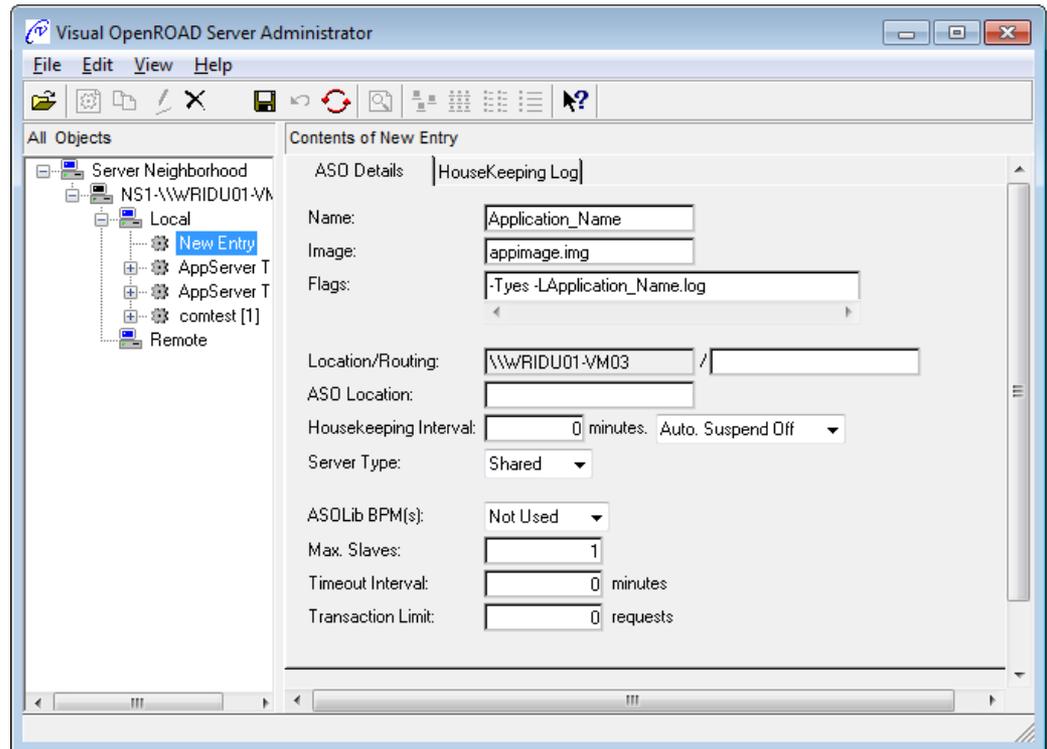
Define Loadrun to the OpenROAD Server

- Use the Register option to define Loadrun



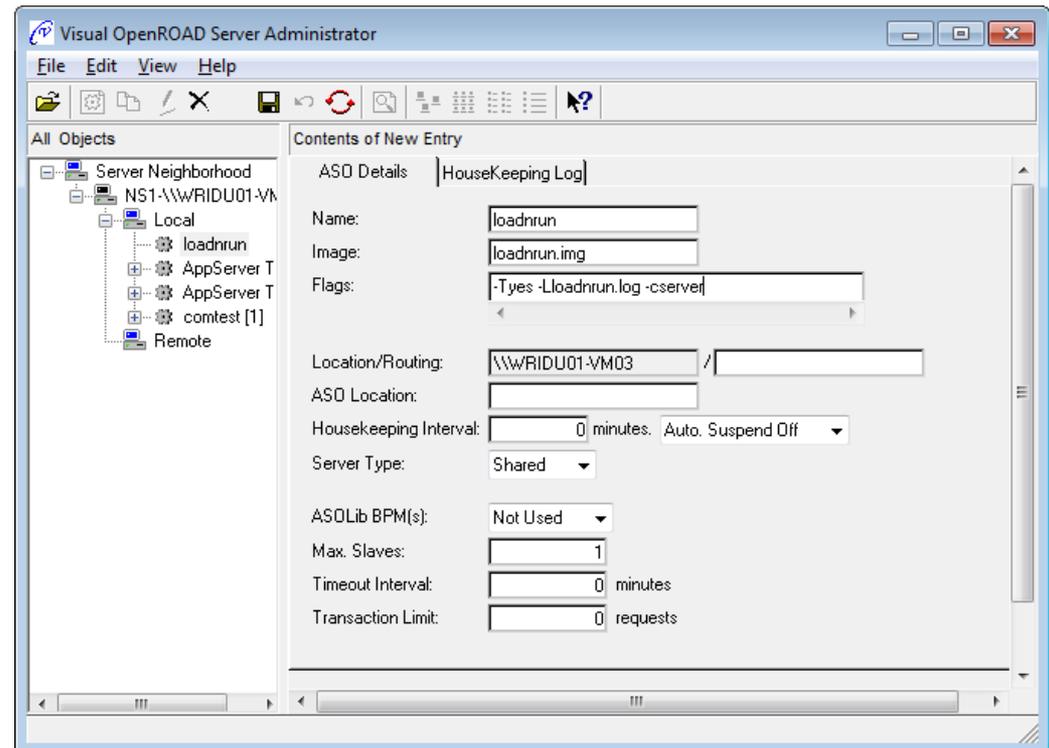
Initial Register Frame

- ➔ The frame to the right appears when starting the Register operation for Loadrun



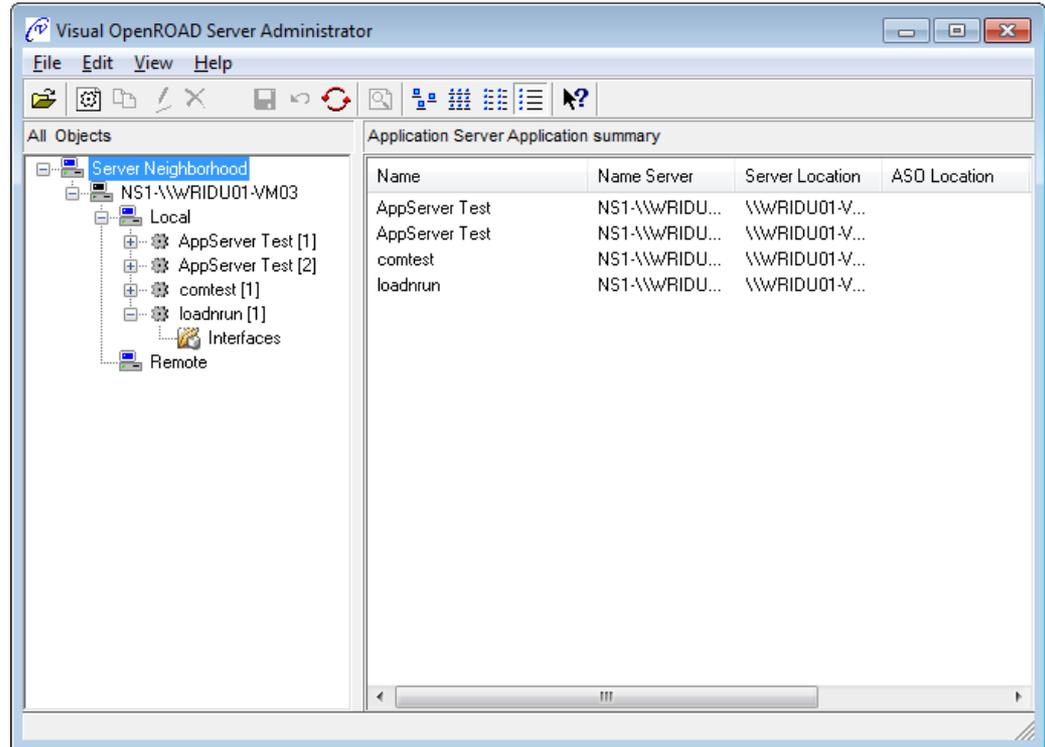
Enter the Loadrun Information

- ➔ It is assumed in this example that II_W4GLAPPS_DIR points to the location that has the loadrun.img file.



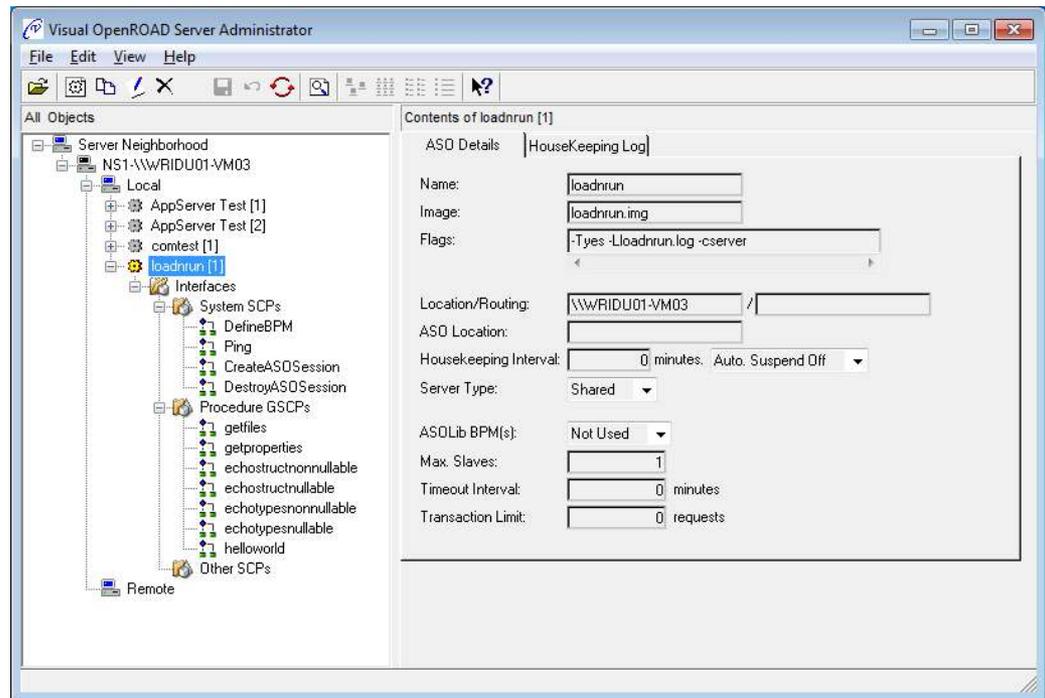
Save the Loadrun Registration

- When the Loadrun registration information is saved, the frame to the right will appear



Verify Loadrun Application is Registered

- ➔ VOSA should be able to access and display the Loadrun application SCPs



Use comtestrso to access Loadrun

- The comtestrso application can be used to access Loadrun via the Gatekeeper.

```
Administrator: OpenROAD Command Window - II_SYSTEM=C:\Program Files (x86)\Ingres\or62xc
C:\Users\wridu01>comtestrso http://localhost/openroad/OperROADGatekeeper.aspx http loadrun
Successfully created component.
Using interface IRun4GLrs.

=====
ImageFile: "loadrun"
CmdFlags: "-Tall,logonly -Lcomtest.log"
Location: "http://localhost/openroad/OperROADGatekeeper.aspx"
Routing: "http"
Private: 0
=====

pIRun4GLrs->Initiate; hr = 0x00000000
Input string is: Hello, World.
Input count is: 99

pIRun4GLrs->Call4GL; hr = 0x00000000
Output string is: Well "Hello, World." to you too.
Output count is: 100

Releasing interface IRun4GLrs.
C:\Users\wridu01>
```

Conclusion

➔ OpenROAD 6.2 -New Client-Server and Multi-Tier Deployment

- Features illustrated in the presentation will require the first OpenROAD 6.2 patch
 - p14746 or later
 - Loadrun 6.2 Installer
 - Loadrun 6.0 Installer
 - Loadrun 5.1 installer
- This was the second of three presentations covering OpenROAD 6.2



- See: http://community.actian.com/wiki/LoadnRun_Home

Thank you

Durwin Wright

OpenROAD Engineering

durwin.wright@actian.com

http://community.actian.com/wiki/LoadnRun_Home

