









# The Tools and Machinery Behind the Curtain

Jan Van Bruaene VP of Engineering, Real-Time Innovations





©2018 Real-Time Innovations, Inc. Confidential.









## Our customer's mission critical systems demand high quality products





















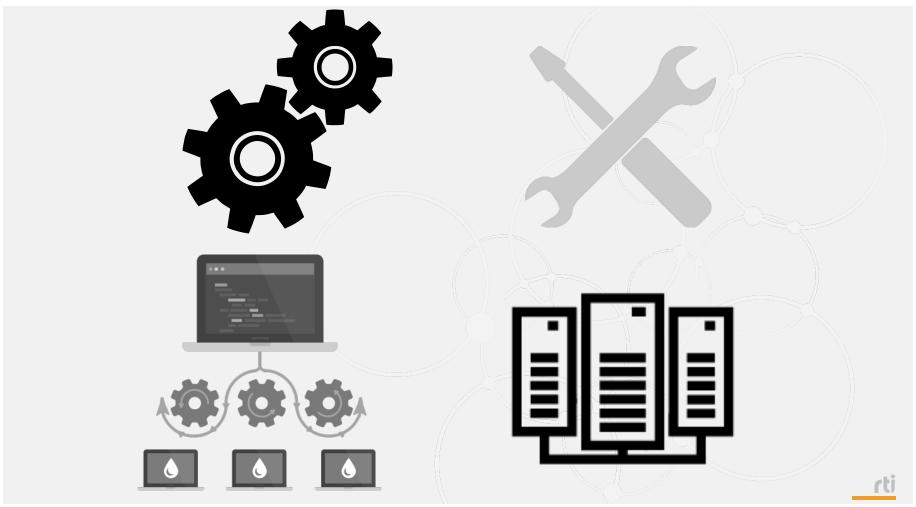












©2018 Real-Time Innovations, Inc. Confidential.

## RTI Quality Management System



#### **RTI Quality Manual and Management System**

Document ID: RTI-000-QualityManual Revision Date: 20180322



opyright 2018 Real-Time Innovations, Inc. (RTI). All Rights reserved. Company Confide

•



## RTI Quality Manual and Management System

Document ID: RTI-000-QualityManual Version 3.x.01 March 22, 2018

Real-Time Innovations, Inc

Page 1 of 45

- Development Process Manual
- Naming conventions
- Document Organization
- Configuration Management
- · Branching and versioning
- Porting process
- Coding guides
- Design and code reviews
- Testing process
- Documentation process
- Release process
- · Defect management
- Root cause and corrective actions
- Maintenance, Support and Escalation processes
- IP, legal and export process
- Audit process
- Vulnerability management process
- Etc.





Customers can sign up to the security notification list by emailing security@rti.com

## **Software Vulnerability Management Process**

#### Document ID: RTI-DEVPo18

Version 1.0

December, 2017

This document captures the guidelines to manage vulnerabilities in the RTI Software Products.

#### Why a software Vulnerability Management Process?

#### When is a bug considered a vulnerability?

Nomenclature

#### Internal tracking of vulnerabilities and vulnerability related information

Jira

How to Track & Document Individual Vulnerabilities

How to Group Set of Vulnerabilities

Salesforce

Email list

#### Fixing vulnerabilities

#### Documenting vulnerability fixes

Release notes

Critical issues report

Security Notice or Notification

Internal-only

Security Vulnerabilities - Technical summary

Security Notification Coordination Document

Security Notification FAQ

#### Vulnerability related patches

#### Sharing information about vulnerabilities

When do we share vulnerabilities with customers?

Review and approvals

Internal Communication

External Communication

Copyright 2017 Real-Time Innovations, Inc. (RTI). All Rights Reserved. Company Confidential.





Pages

### **SEI CERT Coding Standards**

Created by Admin, last modified by Barbara White on Jan 04, 2018

#### **MAINTENANCE NOTICE**

All planned maintenance on the Secure Coding wiki is complete. More information can be found here.

#### Welcome

This site supports the development of coding standards for commonly used programming languages such as C, C++, Java, and Perl, and the Android™ platform. These standards are developed through a broad-based community effort by members of the software development and software security communities.

For more information about this project and to see tips on how to contribute, please see the Development Guidelines.

#### **Downloads**



The SEI CERT C Coding Standard, 2016 Edition (errata)

**Standards Development Area** 



The SEI CERT C++ Coding Standard, 2016 Edition (errata)

#### News

December 2017: The Fall 2017 Edition of the Secure Coding newsletter was published on 20 December 2017

2 Log in

November 2017: Lori Flynn and Will Klieber authored a blog post on Automated Detection of Information Leaks in Mobile Devices.

April 2017: David Svoboda writes in the SEI Blog about the CERT C++ Secure Coding Guidelines.

March 2017: The SEI CERT C++ Coding Standard (2016 Edition) is released.

September 2016: Aaron



**Շonfluence** Spaces ▼

Q ? Log in Sign

- MEM02-C. Immediately cas...
- MEM03-C. Clear sensitive i...
- MEM04-C. Beware of zero-l...
- MEM05-C. Avoid large st...
- MEM06-C. Ensure that sens...
- MEM07-C. Ensure that the ...
- MEM10-C. Define and use ...
- MEM11-C. Do not assume i...
- MEM12-C. Consider using ...
- > Rec. 09. Input Output (FIO)
- > Rec. 10. Environment (ENV)
- > Rec. 11. Signals (SIG)
- > Rec. 12. Error Handling (ERR)
- > Rec. 13. Application Program...
- > Rec. 14. Concurrency (CON)
- > Rec. 48. Miscellaneous (MSC)
- > Rec. 50. POSIX (POS)
- > Rec. 51. Microsoft Windows (...
- > 4 Back Matter
- > 5 Admin
- Space tools

Pages /... / Rec. 08. Memory Management (MEM)

### MEM05-C. Avoid large stack allocations

Created by rlane, last modified by Svyatoslav Razmyslov on Mar 12, 2018

#### Risk Assessment

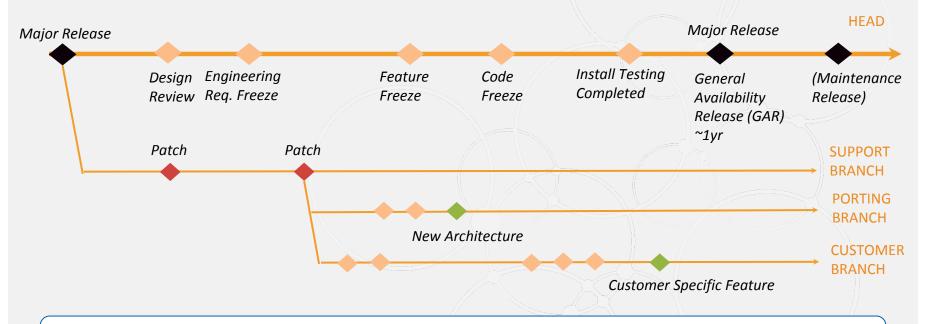
Program stacks are frequently used for convenient temporary storage because allocated memory is automatically freed when the function returns. Generally, the operating system grows the stack as needed. However, growing the stack can fail because of a lack of memory or a collision with other allocated areas of the address space (depending on the architecture). When the stack is exhausted, the operating system can terminate the program abnormally. This behavior can be exploited, and an attacker can cause a denial-of-service attack if he or she can control or influence the amount of stack memory allocated.

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
MEM05-C	Medium	Likely	Medium	P12	L1

#### **Automated Detection**

Tool	Version	Checker	Description
CodeSonar	4.5p1	IO.TAINT.SIZE	Tainted Allocation Size
		MISC.MEM.SIZE.BAD	Unreasonable Size Argument
Coverity	2017.07	STACK_USE	Can help detect single stack allocations that are dangerously large, although it will not detect excessive stack use resulting from recursion
Klocwork	2017	MISRA.FUNC.RECUR	
LDRA tool suite	9.7.1	44 S	Enhanced Enforcement
Polyspace Bug Finder	R2016a	Tainted size of variable length array	Size of the variable-length array (VLA) is from an unsecure source and may be zero, negative, or too large Size of variable-length array is zero or negative
		Variable length array with	

### Documented development processes and best practices Development Milestones



- Each release can include new features, new RTI Connext products, architecture ports and bug fixes.
- Certain features are listed as 'experimental' or products may be 'Beta' releases

Note: not shown are release branches

## The Tools

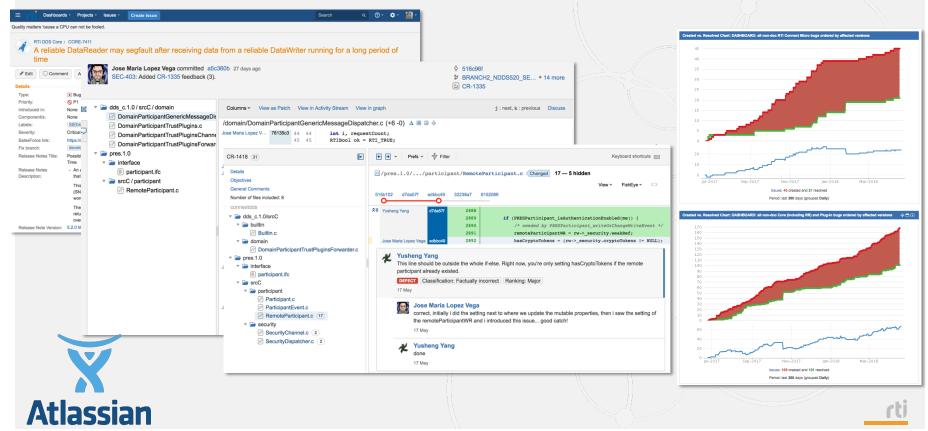


rti

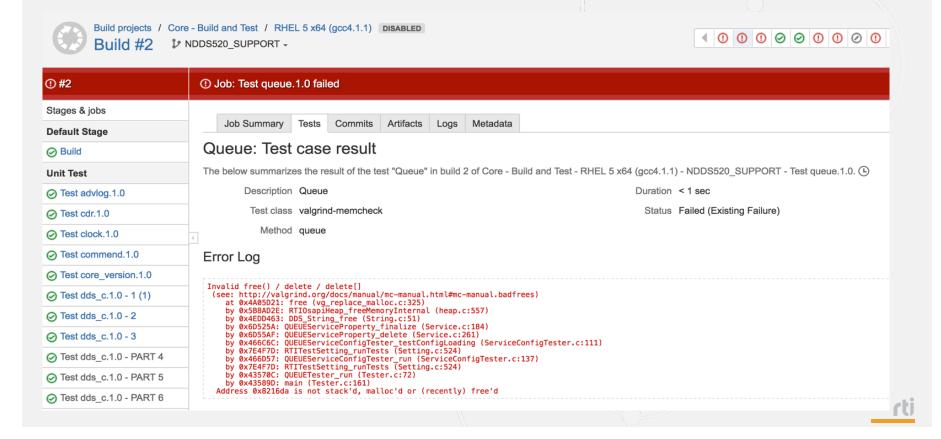
- Configuration management: Git; Atlassian Fisheye
- Defect management and traceability: Atlassian JIRA
- Enhancement management: Atlassian JIRA (and Salesforce.com)
- Customer case tracking: Salesforce.com
- Project tracking and collaboration: Atlassian Jira, Atlassian Crucible, Software
   Development Folders in Egnyte, Trello, Microsoft Project and Excel
- IDE: Visual Studio, Eclipse, SlickEdit, Xcode, atom, zerobrane, Wireframe Sketcher
- Continuous Integration: Atlassian Bamboo
- Memory tools: MemScope, Rational Purify, Instruments (Mac), LeadDiag, drmemory,
   Valgrind and JVisualVM
- Code coverage tools: Rational PureCoverage, gcov
- CPU profilers: Callgrind, Rational Quantify, JProfiler
- Static Analysis tools: FindBugs (Java), clang and cppcheck
- Installers: BitRock InstallBuilder

Work in progress: Atlassian BitBucket, Jenkins Cl

## Atlassian tools provides us full traceability



## Dynamic Analysis using Valgrind with/without Buffer Pools



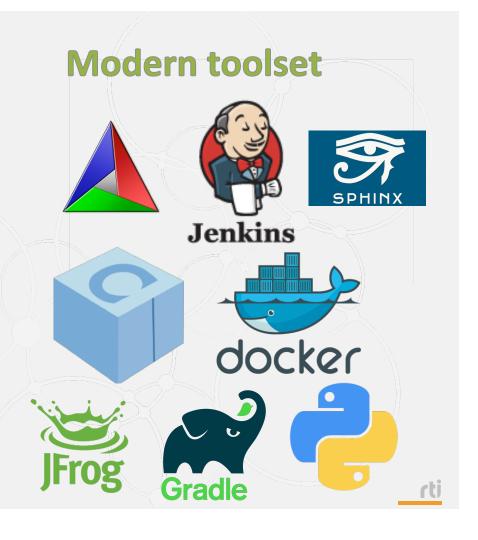
### **Current toolset**

**Gmake** 

Cygwin

**NFS** 

Older and non-native versions of various tools



## Testing

**Regression testing** 

Feature testing

Interoperability testing

Install testing

Performance and scalability testing

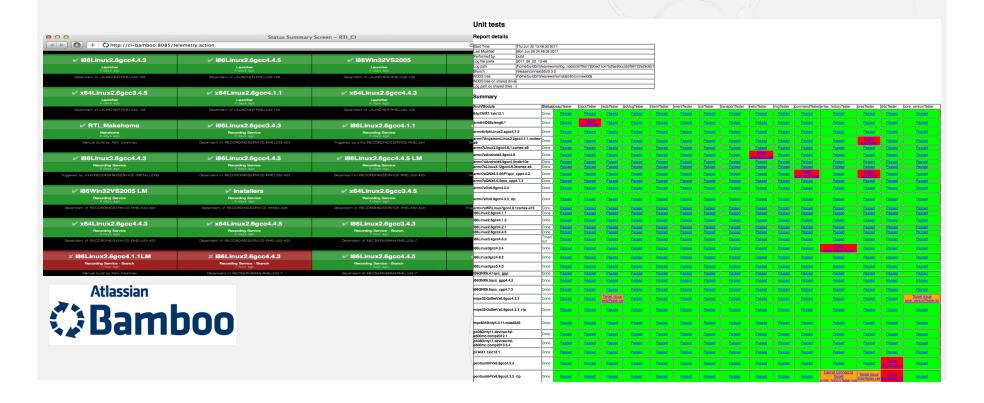
Security testing

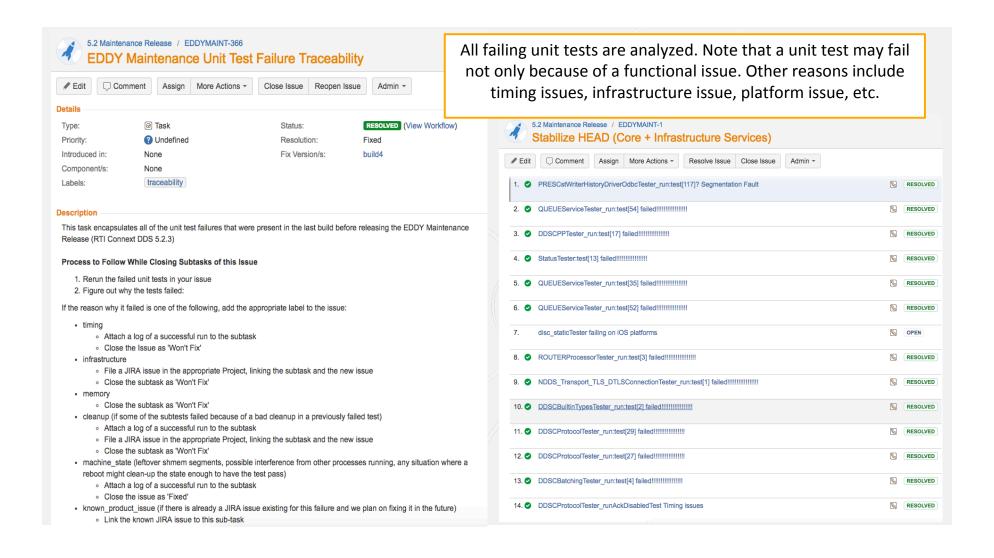
**Endurance testing** 



## Unit tests - Regression testing

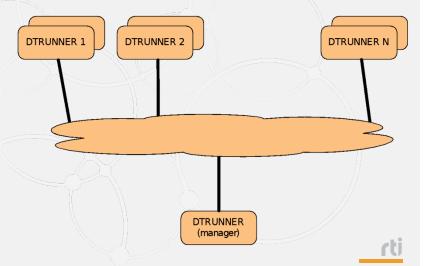
Also provide a level of single node feature testing





## Feature testing

- Our previous automated **RTI Connext DDS** distributed feature framework has been phased out and will be replaced by a new automated distributed feature test framework.
  - Local node feature testing is covered partially within unit tests.
  - Manual feature testing
- RTI Connext Micro has a new distributed feature test framework and test description language.



## Interoperability testing

- Vendor interoperability testing done at OMG events through shapes demo.
- RTI Connext product interoperability testing is currently done manually
  - During install testing of various services and tools, during development, and during release testing of Micro.
  - Manual testing with older RTI versions is done, when after analysis, we determine there is a risk of break interoperability.
- Language interoperability testing is done indirectly through the usage of tools like admin console (written in Java) in combination with applications in other languages. We have also added automated regression tests for specific features (e.g., keyhash).
- Note: Interoperability issues are documented in the release notes, including backward interoperability options.

## Install testing captures integration and interop issues

### Manual install tests plan procedures cover

- Installation are all files properly installed?
- Graphical User Interface tests, e.g.,
  - Integration RTI Launcher with Prototyper, with rtiddsgen,
  - Admin Console
- Documentation tests
- Minimal functionality tests for all products using the shipped examples. For some products, we run a full functionality test using the Getting Started Guide. This is performed on a variety of platforms.
- Product and language interoperability testing (limited)

#### DO NOT MODIFY

This is now an archived version of the document which is specific to the Eddy delivery (5.2.0).

## Eddy Manual Install Test Procedures

#### Introduction

This document contains the manual install test procedures for each RTI product as well as general instructions you should read. It is intended to provide the basic tests common to any release. Of course, each release is different and it is expected that each release will add steps to satisfy its particular needs.

This document is work in progress, so please provide feedback to improve it and make testing easier for everybody.

#### How to read this document?

- Fully read the "General instructions" section.
- After that, find the instructions for the product you are testing.

Introduction
General instructions
Procedures

Test assignment and feedback Filing and resolving issues

Domain IDs Resources

> Testing machines Users

Bundles and installers Licenses

Install Test Procedures 1



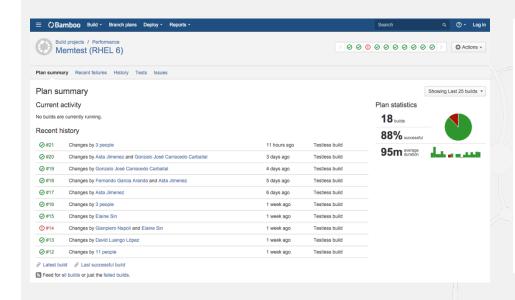
## Automated Install testing

- Installation filecheck to make sure files are properly installed.
- Running rtiddsping, rtiddsspy and prototyper.
- Running rtiddsgen generated examples in C, C++, C++03. C++11, C++ CLI, C#, Java, using a combination of static/dynamic and release/debug DDS libraries
- Running shipped examples using a combination of static/dynamic and release/debug DDS libraries
- Performance examples in C++ and Java
- TCP shipped examples in C
- On more than 80 different architectures including Windows, Linux, Solaris, Lynx, QNX, Darwin, VxWorks platforms. Not covered on e.g., VxWorks 653, etc.

## Performance and memory profiling tests

- Unit tests capture performance and memory information for specific functions
- A bespoke performance test (PerfTest) is used to characterize the performance of Connext DDS, and can be used in conjunction with other products, e.g., Routing Service. (Available on community.rti.com)
- A specific memTest was created to monitor the memory footprint of Connext DDS.
- Connext Micro captures memory information through its test framework
- Admin Console takes performance snapshots of performance critical functions.
   The results are analyzed over time
- Continuous integration of PerfTest and MemTest tests ensures we do not regress as new features are added to the Connext DDS product.

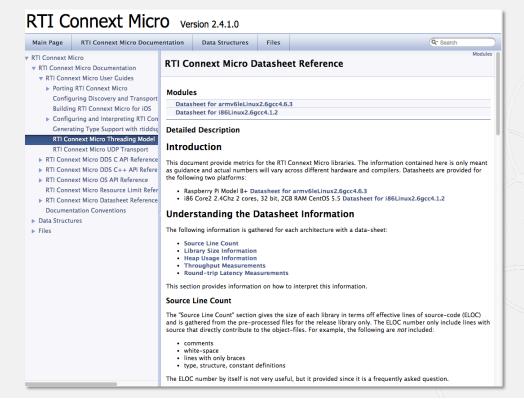
Automated performance, memory profiling and stress tests compare variations with previous revisions. Performance data is gathered at all levels of the product

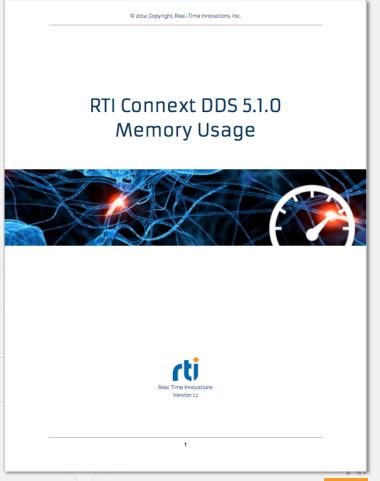


I86LINUX2.6G	CC4.4.5	MEMORY	CONSU	MPTION R	EPOR <sup>-</sup>
Entity	New HEAD	Base (GAR)	Diff (Bytes)	Diff (percent)	OK/ERR
Participant	912347	912012	335	0	ок
Туре	1009	1009	0	0	ок
Topic	1321	1321	0	0	ок
Subscriber	9012	9012	0	0	ок
Subscriber	2196	2196	0	0	ок
DataReader	49788	49788	0	0	ок
DataWriter	27984	27984	0	0	ок
Instances	276	276	0	0	ОК
Sample	845	845	0	0	ок
Remote DR	3853	3542	311	8	ОК
Remote DW	8539	8538	1	0	ок
Instances DR	629	800	-171	-21	ОК
Samples DR	627	800	-173	-21	OK
Remote Participant	31120	30547	573	1	OK



## Automated Memory Usage Statistics





## **DDS** Secure Testing

- DDS Secure functionality testing, including negative testing. E.g.,
  - configure DDS Secure with invalid governance or permissions file
  - configure DDS Secure with configuration files, with duplicate grant or subject names.
  - muck with the clock (e.g., provide a file with date past 2038)
  - tamper with wire communication
  - enable DDS Secure plugin with missing functions
- DDS feature interoperability testing validating how security interoperates with various DDS features and products (e.g., reliability, batching, built-in types, CFT, different transports, routing service, persistence service, etc)
- DDS Vendor interoperability testing
- Robustness and Performance testing
  - Valgrind (with/without RTI buffer mgt), perftest, discovery scalability
- · Usability testing
- Red team testing at DoD IA Range in Quantico VA, part of the USS SECURE project
- Fuzz testing

## **Endurance Testing**

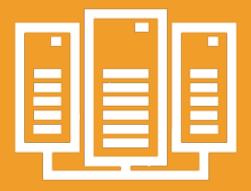
Testing for memory growth, or other unexpected scenarios in long running scenarios:

- 1. Start publishing samples using a high sequence number.
- 2. Remote participants appearing and disappearing.
- 3. Remote endpoints appearing and disappearing.
- 4. Endpoints communicating with different instances and message sizes, including batching.

More to come.

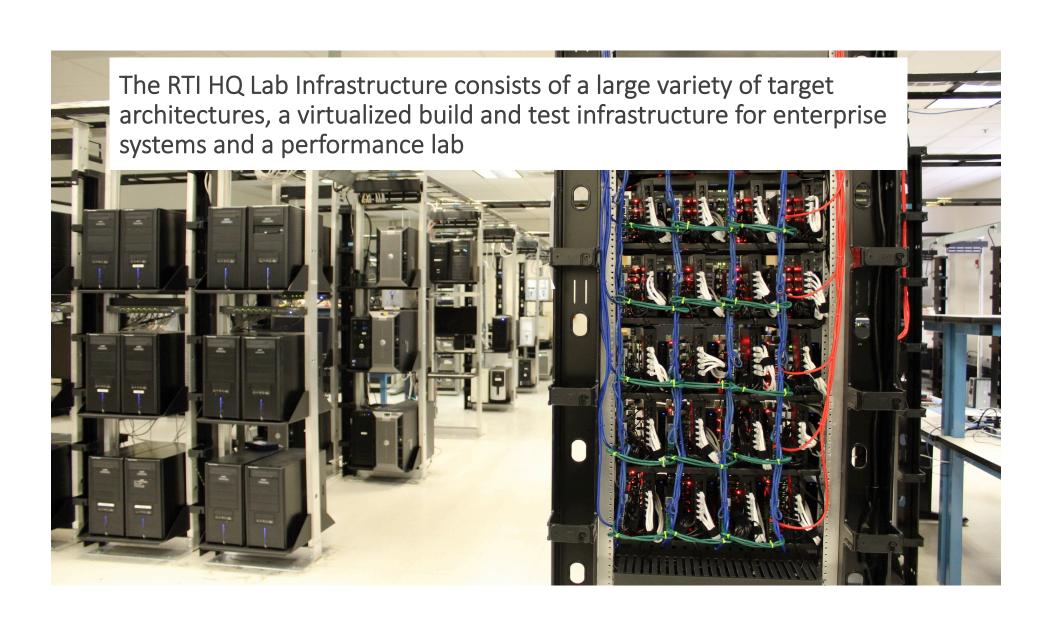


## The RTI IIoT Lab



## Our products are supported on a large set of architectures

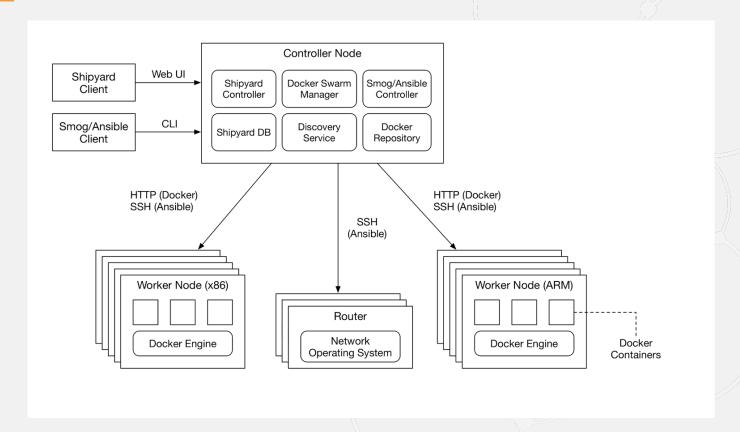
Mind Rear Persons II P	es 6, server 2012 R2	CPU Architecture. ddi ddi ddi ddi ddi ddi ddi ddi ddi dd	VS 2017 VS 2017 VS 2015 VS 2015 VS 2015	Library*  4649/Linux 8,004.4.1  699/6029207  4649460397  469960397				IDPI Multi- TC PV6 cast IPI			Monitoring Library 0							rices			Spreadsheet Add-in	Tools							Data Integration Services			Ada Binding (available and of Assa	Ounies Limites	Co	CORBA Competts II
Mind Rear Persons II P	Dev Chrus 6 sis 13, server 2016 sis 13, server 2016 sis 13, Server 2016 sis 13, Server 2016 sis 13, Server 2017 for 15, Server		VS 2017 VS 2017 VS 2015 VS 2015 VS 2015	664WC:inu2 5god 4.1 66Wn32V52017 66Wn32V52017 86Wn32V52015							Monitoring Library 0		Secure WAN								Add-in							Country					Osmira United	Bandwidth	
Mind Rear Persons II P	Dev Chrus 6 sis 13, server 2016 sis 13, server 2016 sis 13, Server 2016 sis 13, Server 2016 sis 13, Server 2017 for 15, Server		VS 2017 VS 2017 VS 2015 VS 2015 VS 2015	664WC:inu2 5god 4.1 66Wn32V52017 66Wn32V52017 86Wn32V52015							Library 0										(Compatible with MS Office Excel														
Mindows Printers II Printers I	es 13, server 2016 es 13, server 2016 es 13, server 2016 es 13, server 2016 es 13, server 2012 FC, Server 2018 es 5, 6.7. es 6, server 2012 R2 es 5, server 2012 R2 es 5, server 2012 R2 es 7, server 2012 R2	504 605 605 605 604 606 604 606 606 606 606 606 606 606	VS 2017 VS 2017 VS 2015 VS 2015 VS 2015	88Win32V32017 #6FWin8VS2317 88Win32V32015												uting Service "	Adapter SOK	Service	Service	Shapes Domo	2210, 2313, 2310)	Launcher	Monitor	Admin Console	Security Plugina	Security Plusina SOS	Oracle Times Ten <sup>17</sup>	Database 11g"	591	PostgreSQL	MySQL"	2017)	Service Ha	nins × ACE	.100 J
Nindows 1 Nindows 1 Nindows 1 Nindows 1 Nindows 1 Nindows 1 Nindows 2 Nindow	ss 13, Server 2016 ss 13, server 2012 FG, Berver 2016 es 8, 8-1 ss 5, server 2012 R2 ss 5 es 8, server 2012 R2 ss 7	664 565 564 569 560 560 684	VS 2513	88Win32/32015		•44				-		-			•21																				-
Windows Divideos B Div	sis 13, server 2012 RQ, Server 2016 sis 6, 8.1 sis 6, server 2012 R2 sis 5 sis 8, server 2012 R2 sis 7	x64 x68 x64 x65 x64	VS 2513			- 44		25 25 28	: :	-:-	-:-	÷			24	-:	-:	-:-	-:-	-:	- :	-:-	-:-	-:		- :			_	_	_	_	•	•	_
Directions 8	es 6, server 2012 R2 es 6 es 6, server 2012 R2 es 7	691 694 694	VS 2013	x64Wn84V52315		•**		26		•	•				24	:			:					-:	•	-:					•"			:	
OS X 10 15 15 15 15 15 15 15 15 15 15 15 15 15	ns 8, senver 2012 R2 ns 7	986 984	99 2013	88Win02V82013 464Win04V52213	: :	***		25	: :	- :		-:-	:	-:	. 2		:	:	-:		- :	- :	- :		- :						• "				
05 X 50.12 6.11 6.10	es 7 2008 R2, Windows 7		VS 2012 VS 2012	88Win32VS2012 #66Win86VS2012	1: :	- **		20	: :	-:-		-:-	:	:	24	:	:	- :	- :	- :	- :		- :	- :	-:	:		:	- :		. "			:	
05 X 50.12 6.11 6.10	2008 FQ , WINCOWS 7	dt	VS 2510 SP1	88Wik02VI2010 864W164VIS2010				25		•		•		-:-			•	:			-:	-					····							:	
90.10 90.10		64 64	dangkili	#642arwin17clang8.0	-	44		a	• •	-÷		<del>-</del>	-:	$\cdot$	26	-	-		-:-	-:-		-:-	-	-					_		_		•	<del></del>	-
90.10		401 004	dang 8.0	afit2amin1fctangl(0	•	•**	•	- : :	: :	-:	-:		-:	-:	-11	-:-	-:-	-:	-:				-:-	-:	-:	-:-							: -		=
		464		a642arwin14stang8.0	•	**	•			-	•	-		-	21	•	•	-	- i	-:		-:	-	-									•	=	=
Linux (ARM)	NO LTD	ARMA SANS	900 5.4.0	arm/TArgetonLinux3 2gcol 4.1 cotex-a9.11 arm/Linux4.4acd.4.0.11		44	•	. : :	: :	-:		-:	- :	-:	_	:	-:					_			-:-	-:-	-					_			-
7.0		A01	900 4.0.1	perEuroH197.0gook.8.1		. 0		.00	•		. 10	•																							=
6942		PPG (PPG32 W FPU)** PPG (4500-2)	goz 4.3.3 goz 4.3.3	ppx6047x6.9.4gcx4.3.3 ppxs600/27x6.9.4gcx4.3.3			- : :	20	- : -		- :	-:		_ F																_	_	$\perp$			
8932		494	gor 4.3.3	pentumbitivis Sport 3.3 pentumbitivis Sport 3.3				21 21 25 28	-:	•17		•		=	_																			_	=
6.9		PPC (PPC32 w/ FPU) <sup>24</sup>	gor 4.3.3 nor 4.1.2	poct647xt.9aco4.3.3		- :		8	- :	-	:																								
VxWorks 8.8		PPC (PPC)2 w/ FPU) <sup>21</sup>	por 4.1.2	perBustnet.8gcc4.1.2 pod644vi8.8gcc4.1.2																															
4.7		PPC (PPC32 W FPU)**	ger 4.1.2 ger 4.1.2	perfor/Voll.7gos4.1.2 ppd6697x6.7gos4.1.2		- :		25	- :	•17	:																								_
		PPC (PPC32 w/ FPU) <sup>24</sup> PPC 405	gor 4.1.2	pod664/x8.7gco4.1.2_smp pod669/x8.6gco4.1.2				20,27	•																										
68,67		x86 (Pardure)	gor 4.1.2 gor 4.1.2	perdunited agost 5.2			•	•	•	•47	- :			_																_		_			-
63.64		PPC (PPC32 w/ FPU) <sup>24</sup> PPC (PPC32 w/ FPU) <sup>24</sup>	got 4.1.2 got 3.4.4	pod64/x6.6gcx4.12 pod64/x6.3gcx3.4.4		• **	•	•	•		:																								-
		ARMANA SANS	eos 5.4.0	amuliQNOC 0.0quo_gpp6.4.0 <sup>eq</sup> amuliQNOC 0.0quo_quod.4.0 <sup>eq</sup>		•		8 21		•	•	•	-:												-:-	-:									=
7.0			000 5.4.0	#840487.6.0gs;_gsp6.4.0				20		•		•													-:-	- :									
QNX 65 SP1	,	ARMA?	gos 640 gos 442	#660NE7.0 Oper_cost 4.0 arm/TeQNES.5.00Ptocs_cpp4.4.2 **		- :	- : -	25	<del>: :</del>	<del></del>	-:-	•	-:-	-:-											-:-	-:-									$\overline{}$
65		400 400	qur 4.4.2 nor 4.3.3	86QN08.5quc_gpp4.4.2 86QN08.4.1quc_gpp		- :	- :	26	• :	-:	:	-:-	•	•											•	•									=
Android 50.51		460.7s	got 4.9 (NDX r10x)	ams/TaAndroid5.0gcc4.9ndkr13e " ams/TaAndroid2.3gcc4.8 "	•	- 64	•	- : :	:	-:			. 6	-:											-:-	-:-								_	=
108 82		x00 Crusi-Come 642-0 Associa 62	stang ti. 1	486_64C68stang6.1 ann64C68clang6.1		•**	•		•		:	•	• 6	-																					_
Solaris 2.10		UnidPARC	got 3.4.2	sparcSol2.10gcc3.4.2		• **						•		-	. **							_			- :	-							_		-
		UNISPARC POWERZ	gos 3.4.2 XLG/C++ 12.1	sparolitics2.15goc3.4.2 p7A007.1slc12.1	- :			• :						_																_		_	_		-
AIX 2.1		POMER?	XLC/C++ 12.1	64p7ABX7.5xer12.1				- :							•11																				
		p4080	muli 6.1.4	p4080trly11.deutree-fol-e600ms.comp2013.6.4 na080trly11.deutree-fol-e600ms.comp2013.1.1				•			:	•		_																					
Integrity		x86	mult 6.1.4	pentiuminty11 pod6-smp				- :			- :					•17,16																			
10.0.2		×86	nuti 5.0.6	perfunity1102 po86				•			•				•19,21	•17,16																			$\equiv$
50.11		PPC 85xx	multi 4.2.4	ppd86ointy5.0.11.xes-g2020 <sup>32</sup> ppd7400Lyse5.0.0gcd.4.3	$\perp$		- :	-:-	-:-		:			$\overline{}$			-															$\perp$			-2
LynxOS E2		PPC No.	900322	ppcF400Lyns4.2.0pct5.2.2 ppcF400Lyns4.0.0pct5.2.2		· ·		-						_																					
4.0		PPC 354; 7xx PPC 34xx	got 3.2.2 got 3.2.2	ppc/900.ymx4.0.0goc3.2.2 ppc/900Lymx4.0.0goc3.2.2			-		- :	- :																									$\rightarrow$
VxWorks 653 contact PM for 23		PPC 86410	900332	sbc864TV4853-2.3gcc3.3.2 **										$-\top$																					
avoilability)			1		-	_	+		_				$\vdash$	_	_						-	_	-	-					-		-	-	_		-
	Libraries (CTL). Contact Pro	duct Management for av	rallability.																																
64 RedHawk 6.5 Hechtesk i 86 RedHawk 6.5 Rechtesk i	wk Linux 6.5	464	por 4.9.2 por 4.9.2	a64NetHank6.5pcs4.9.2 BBNoPhysis.5pcs4.9.2	-	44	•	: : :	: :	- :		-:	-:	-:-						-:		-	- :	-:	-:-	- :									
4 RedHawk 6.9 RedHawk 1 x86 RHEL 5.2 Red 61.52	ek Linux 60 62	61	por 4.4.5 por 4.2.1	#64,ima2.8gool.4.6 #86,ima2.8gool.2.1						•				-																					=
S00mc Frosscale 1.4 Fromula 1 RWv9 Yooto 2.2 Yests Puty	de 1.4 Nov Trinto (2.2	Quirig Pessonesionesion	gor 4.7.2 por 6.2.0	ppositionest mackgood 7:2 armedi most Secol 2:0 <sup>-0</sup>		44	-:-		- :	-:-	-	-:-		=																				=	=
MN7 XII xx 14.2 Xii xx 14.2	42	ARMAT (Zyng Corsex A6)	got 4.6.1	amin'i mai Signol Si Cortes esi <sup>et</sup>					. :	-				=	_															_				=	=
F Wind River Linux 7 Bird River #MV7 QNX 6.6 QNX 6.5	5	ARMA	gor 4.7.3 gor 4.7.3	arm/TeQNSS.S.Coop. cop4.7.3 <sup>cc</sup>		•**		В				•		:											-:-										=
Werks 6.9.4 dlab 6.9.4.6	5	PPC (MESOC)	600 4.7.3 610 5.9.1	860N08.5ccc pp4.7.3 ppoet509Nd.3.4.8dup5.3.1 **		•"		8		•	50	- :		•				••								•									
0 e590v2 QNX 6.5 QNX 6.5 e600 Integrity 5.0.11 Integrity 5.0	5 y 5 0.11	PPIC (450042) MPIC 8343	por4.42 m/142.4	ppo800020AXE 5.0pp ppp4.4.2 mpo8349mys 0.11.mo8349.71		• "	+ : -		- :-	-	<b>:</b>	•		_	-	. 17	•		_			_					<del></del>	_	_	_	_	+			$\overline{}$





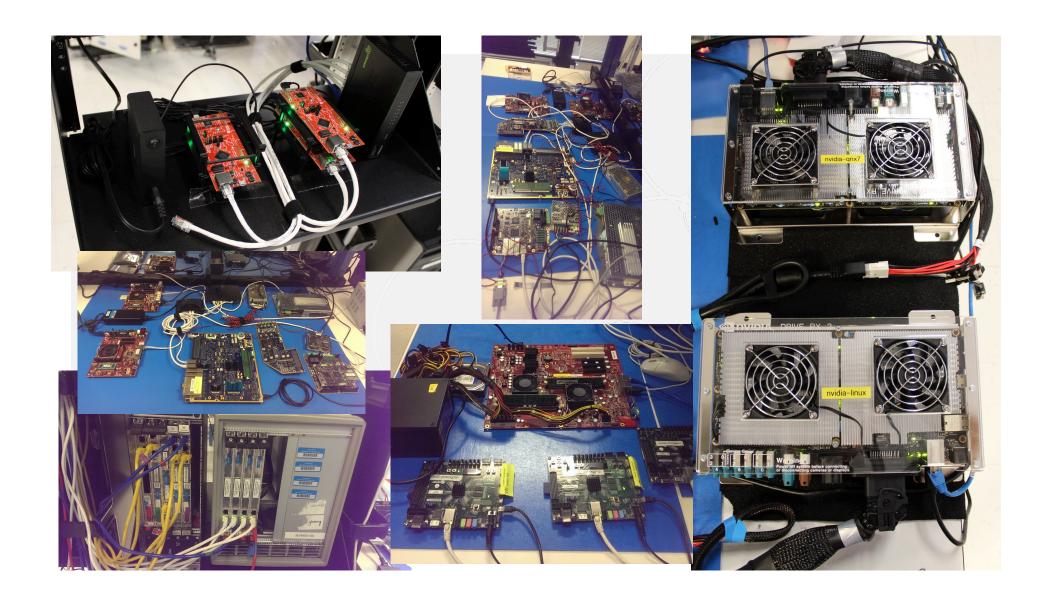
©2018 Real-Time Innovations, Inc. Confidential.

## Software Architecture





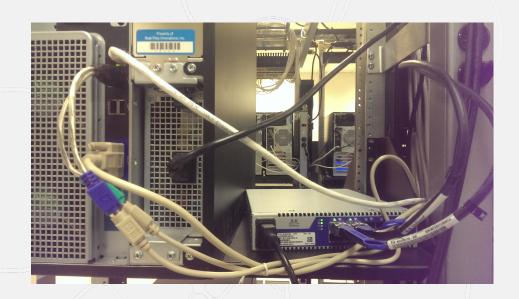
©2018 Real-Time Innovations, Inc. Confidential.





©2018 Real-Time Innovations, Inc. Confidential.

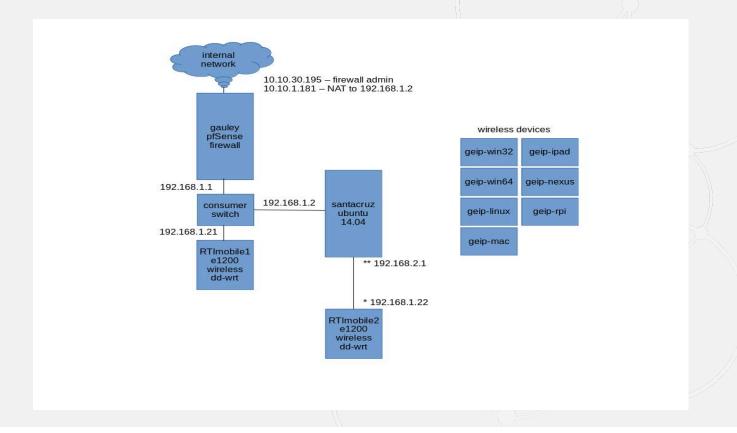






©2018 Real-Time Innovations, Inc. Confidential.

## RTI HQ IP Mobility Test Lab





### Great products are built

... by great and motivated people

... with the right tools

... and guided by the professional processes





## Thank you

