Boeing Intelligence & Analytics

OpenCPI & GNU Radio Integration for Heterogeneous Processing

David Pocratsky
9/14/2017

Copyright © 2017 Boeing. All rights reserved.

Agenda

- Introduction to OpenCPI
- Comparisons to Existing Frameworks
- Motivation for GRC Integration
- OpenCPI and GRC (Demo)
- Road Map for Future Development
- How to Get More Information

Copyright © 2017 Boeing. All rights reserved.

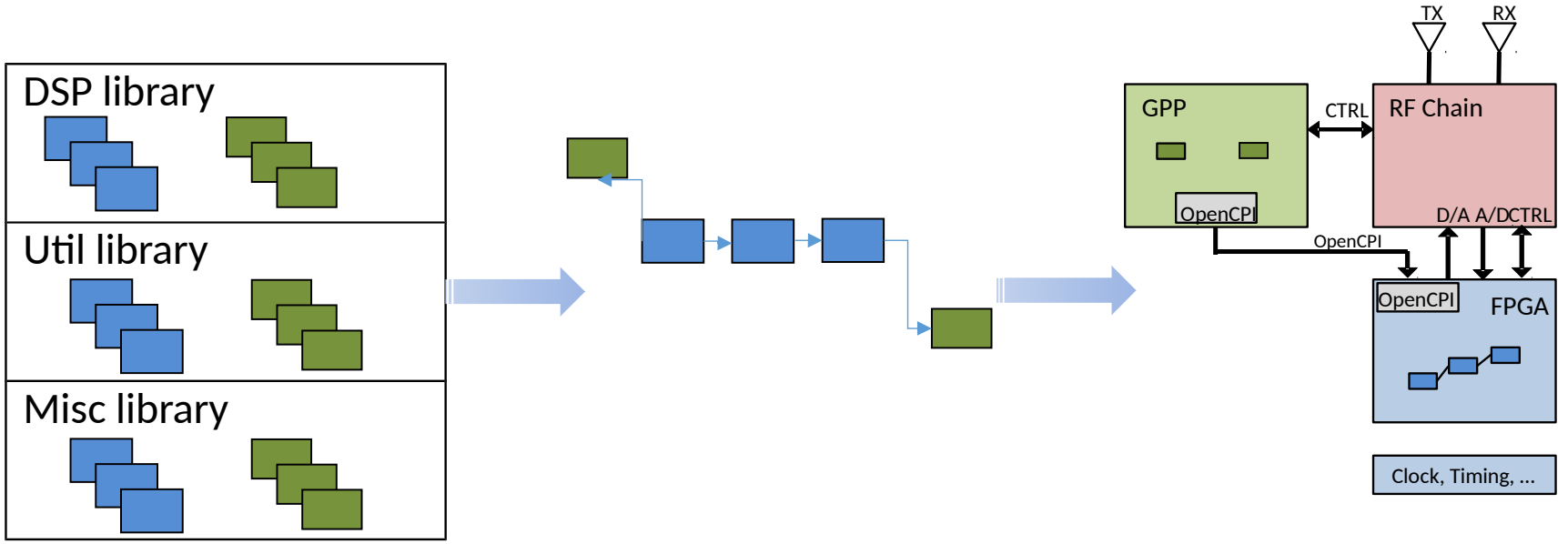
Introduction to OpenCPI

- **O**pen **C**omponent **P**ortability **I**nfrastructure
- Component-based framework for developing portable processing applications targeting various processing technologies
- Processing technologies can include multiple field programmable gate arrays (FPGAs), general purpose processors (GPPs), graphical processing units (GPUs)*, and a variety of interconnection technologies
  - Designed to be platform agnostic and easily adaptable to new platforms
- Integrates existing tools for FPGA development such as ISE, Vivado, XSim/ISim, ModelSim, Quartus II, etc.

*in development/coming soon

Copyright © 2017 Boeing. All rights reserved.

Introduction to OpenCPI



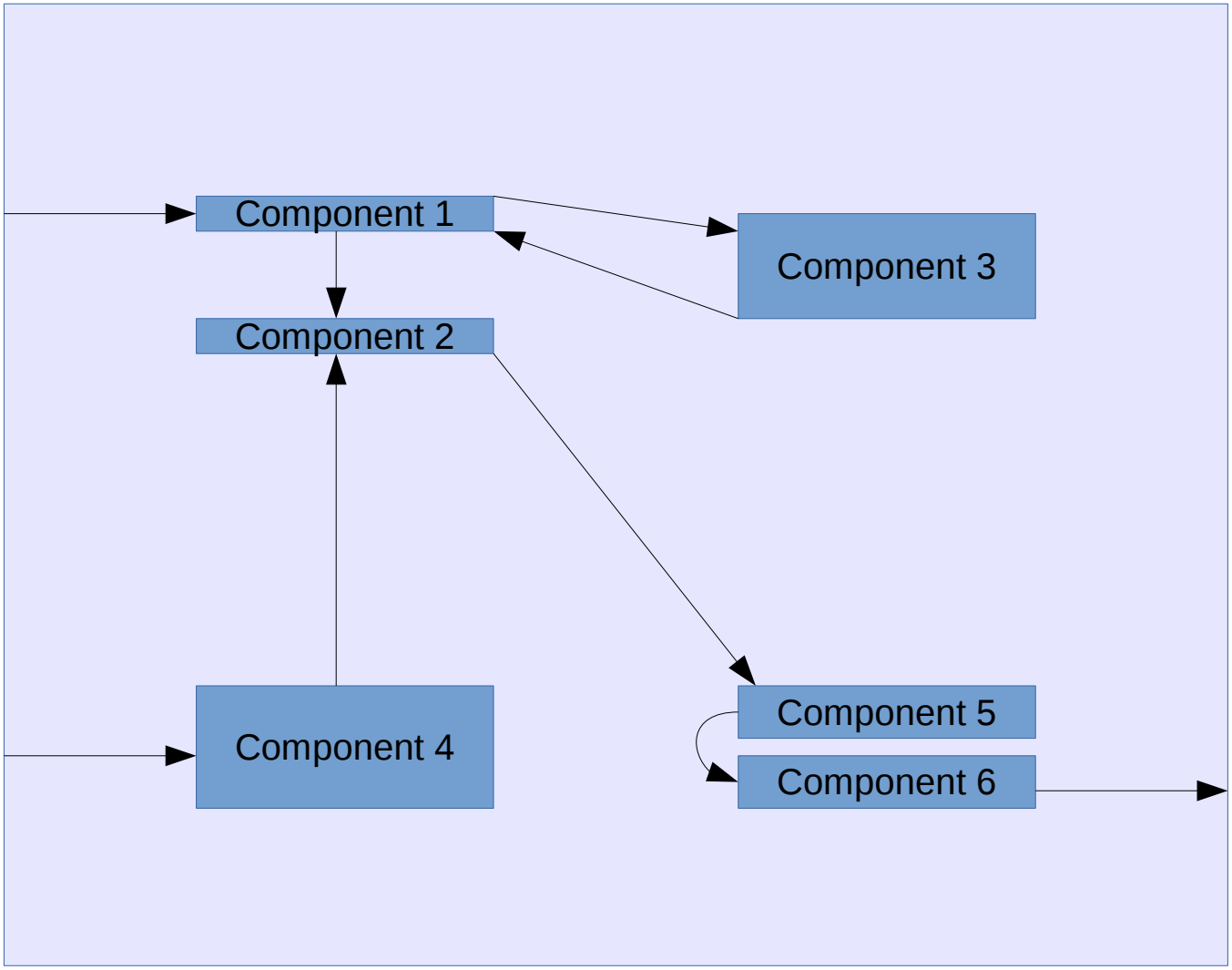
Portable, reusable components are developed in native language for the final target platform (i.e. VHDL/C++) independent of intended application

Applications are constructed from existing components

Application deploys to target platform, with components executing on disparate parts of platform

Copyright © 2017 Boeing. All rights reserved.

OpenCPI – Applications and Components



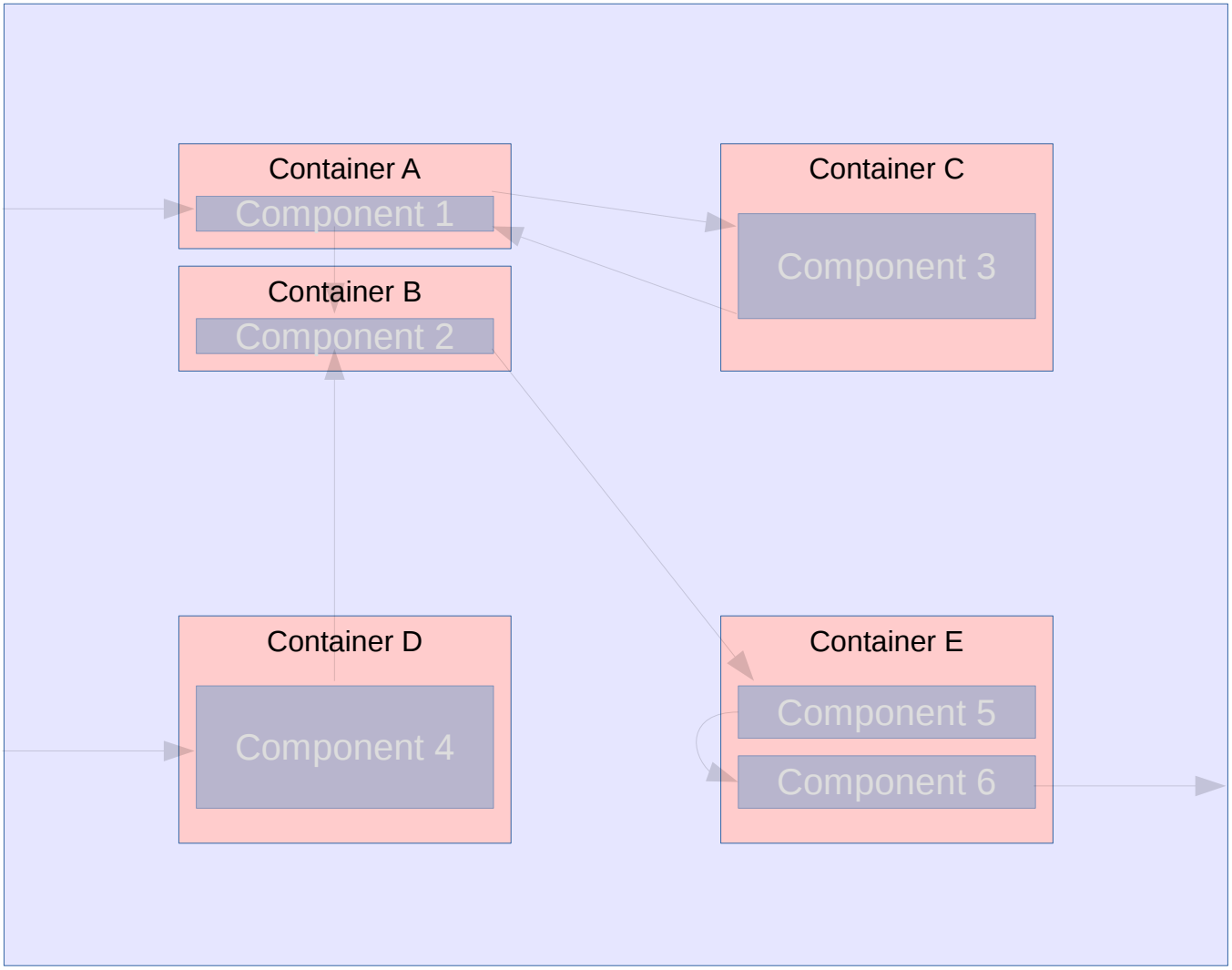
Application – A collection of interconnected component specs

Component – A specification of a function or operation implemented by Workers to run on specific platforms

Worker – A concrete implementation of a component

Copyright © 2017 Boeing. All rights reserved.

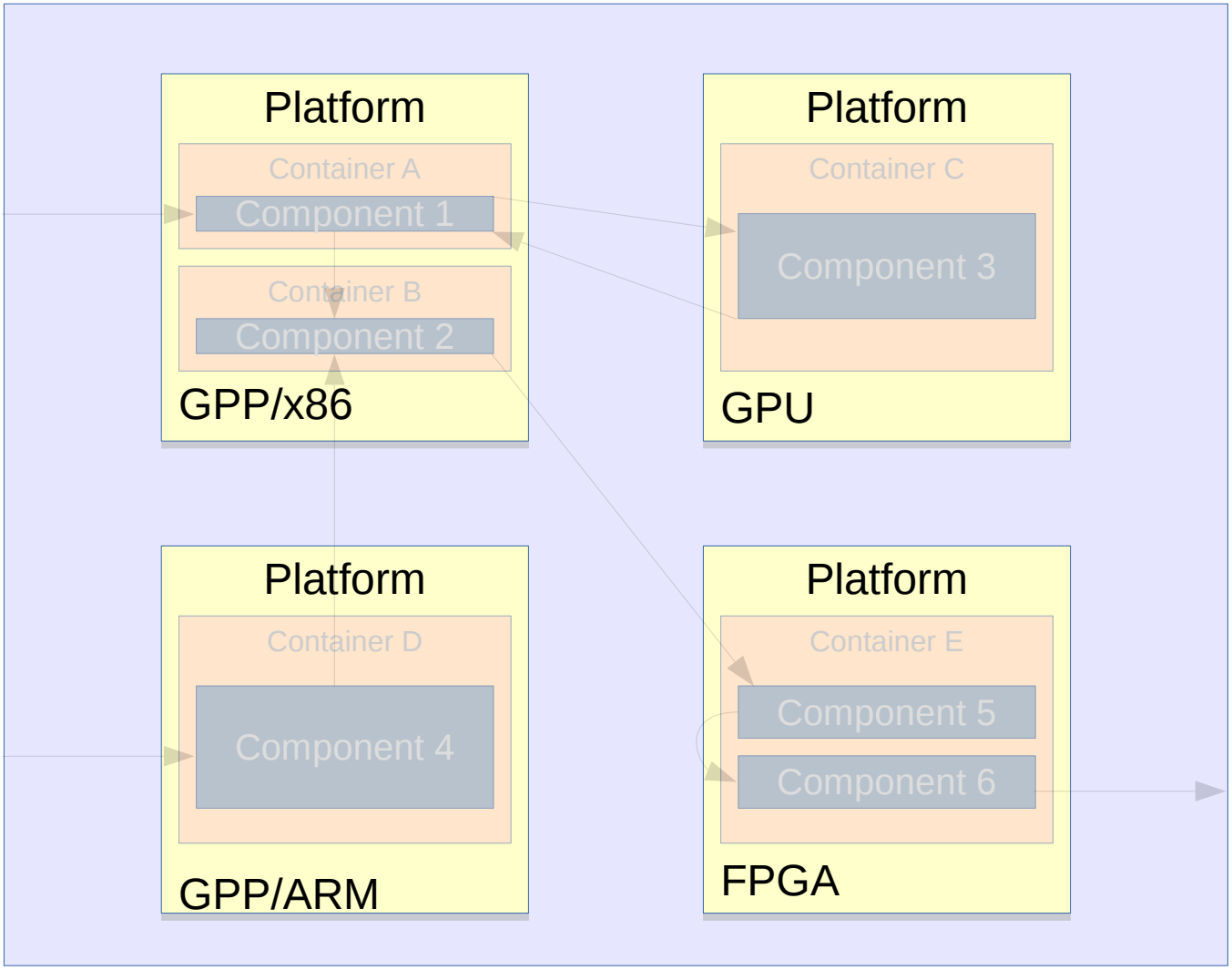
OpenCPI – Containers and Assemblies



Container – Execution environment that runs on some platform that executes workers

Copyright © 2017 Boeing. All rights reserved.

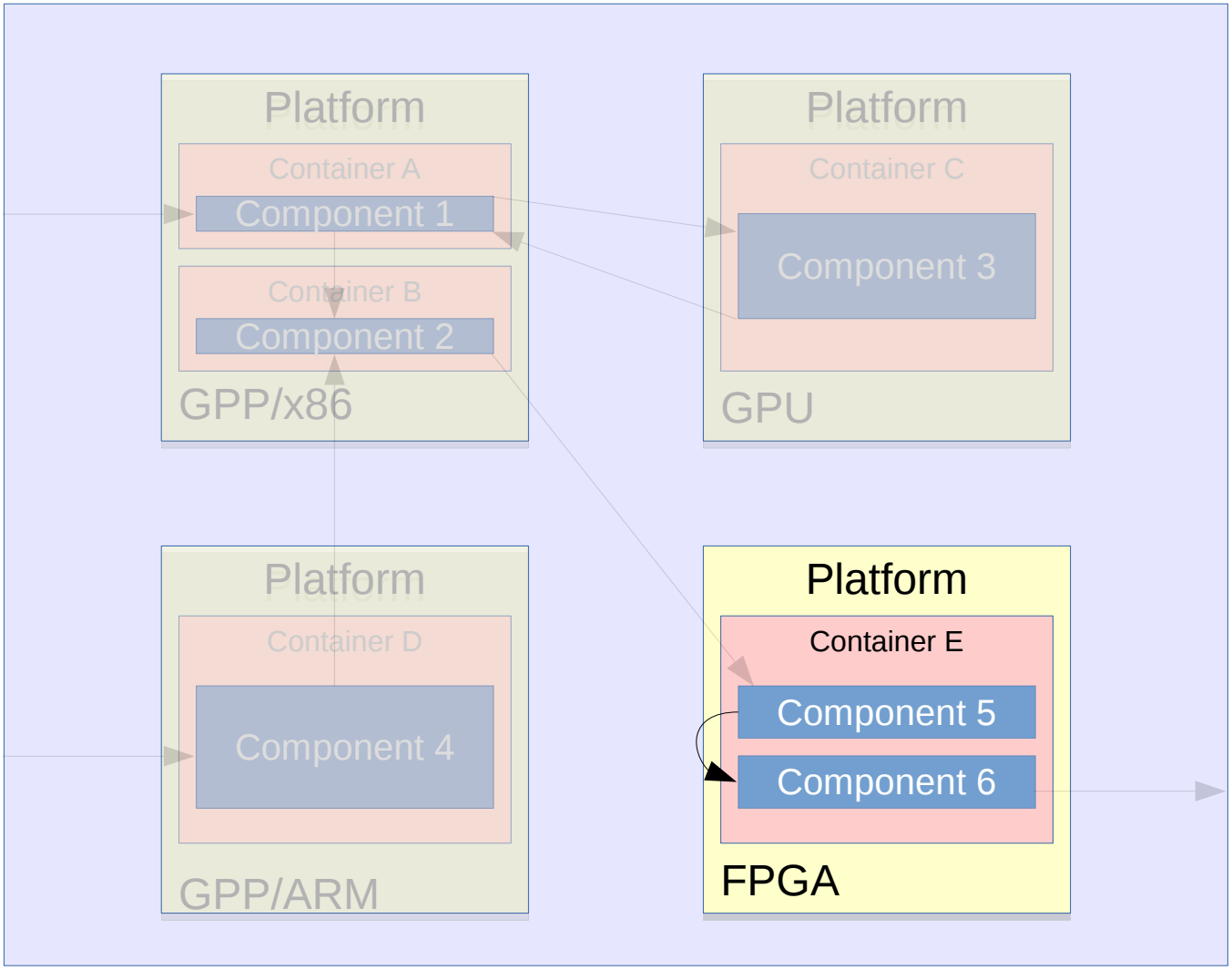
OpenCPI - Platforms



Platform – Physical device which houses one or more interconnected Containers and associated I/O

Copyright © 2017 Boeing. All rights reserved.

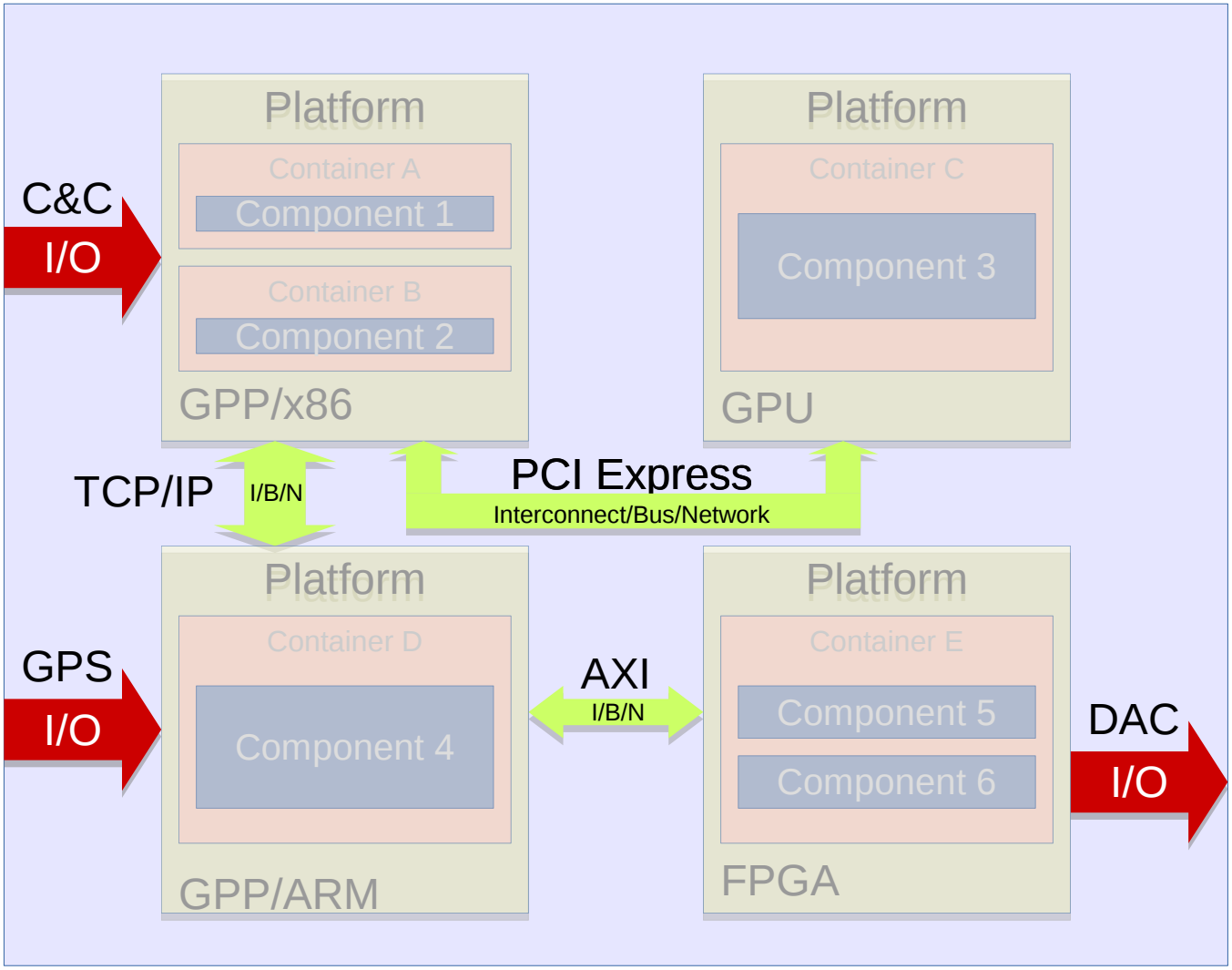
OpenCPI - Platforms



Assembly – Collocated subset of an application built from one or more HDL workers

Copyright © 2017 Boeing. All rights reserved.

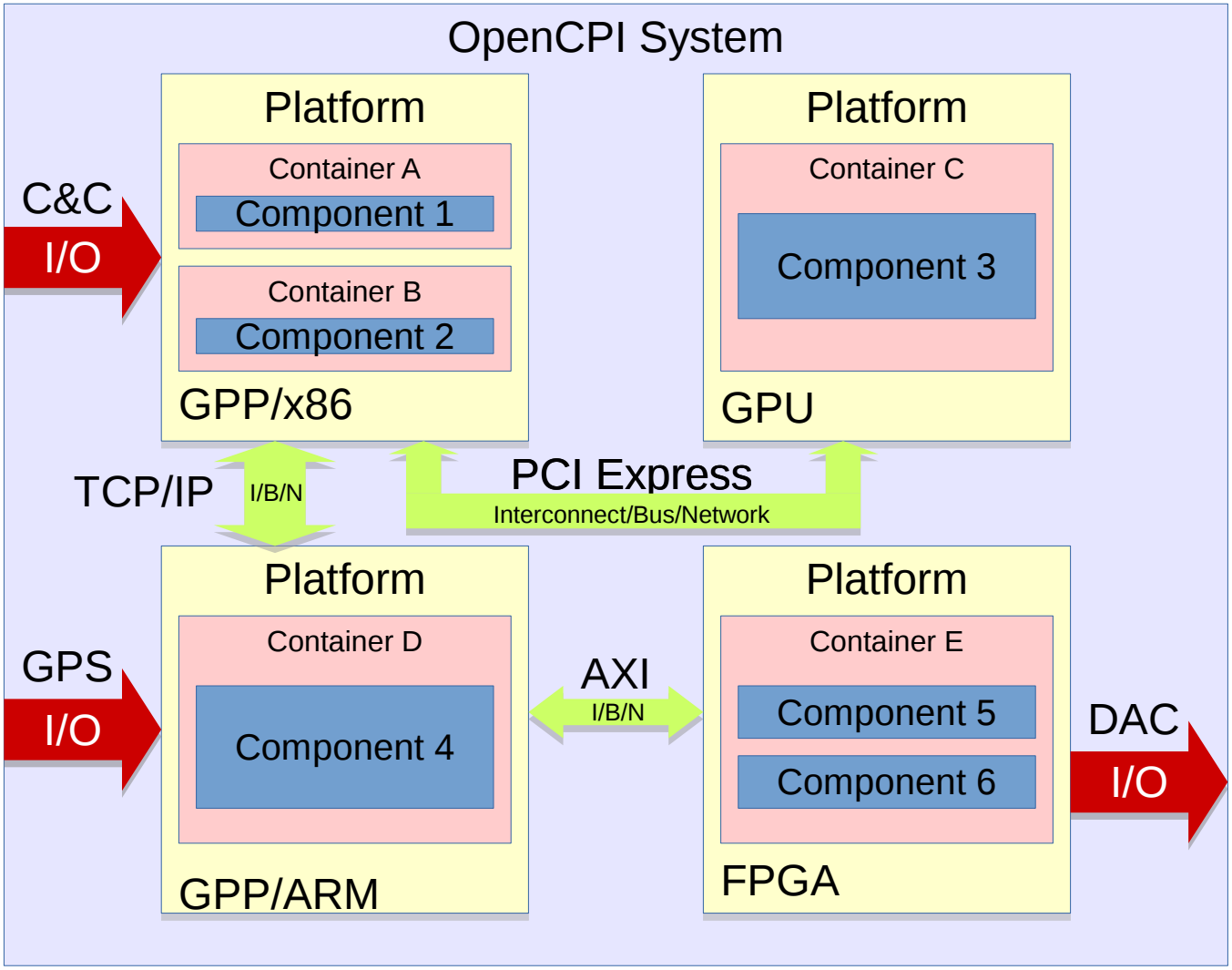
OpenCPI – Interconnect/Bus/Network & I/O



Interconnect/Bus/ Network & I/O – Physical connections between Platforms that is typically abstracted away from the developer

Copyright © 2017 Boeing. All rights reserved.

OpenCPI – OpenCPI System



System – A collection of interconnected OpenCPI-enabled Platforms that can be used to execute Applications

Copyright © 2017 Boeing. All rights reserved.

Comparisons to Existing Frameworks

	GNU Radio	RF NoC	OpenCL	REDHAWK	OpenCPI
Software Support	X	X	X	X	X
FPGA Support		X	Supported with vendor specific extensions	Treats FPGA as black box	X
Access FPGA I/O (ADC/DAC)		X			X
GPU Support			X		Planned for future release
Platform Agnostic			X		X
Leverages Native Platform Language	X	X		X	X
Distributed				X	Support for distributed processing across embedded targets planned for future release

Copyright © 2017 Boeing. All rights reserved.

Motivation for GRC Integration

- Provide a front end GUI for the OpenCPI Application Developer to quickly test, design, and run OpenCPI Applications or Assemblies – Mostly work currently done by hand
- Leverage the existing work that was done in GRC that does not fall within OpenCPI's domain
- Introduce OpenCPI to the GNU Radio community in a way that is familiar and easy to use
- Enhance GRC to be able to run certain blocks on various hardware platforms such as FPGAs, GPPs, GPUs, etc

Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – General Outline

- Install OpenCPI and build the assets for the desired platforms
- (Optional) Write and build your own OpenCPI workers specific to your application
- Run ocpigr to import the workers into GRC as GRC blocks
- Create flowgraph as you normally would with vanilla GRC
- Set the deployment to target the desired containers for each block
- Build/Run the application

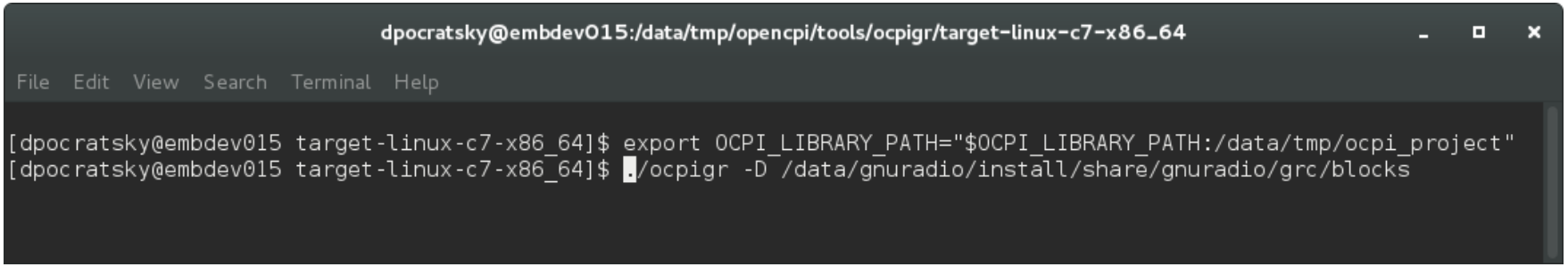
OpenCPI Documentation

Covered in this presentation

Copyright © 2017 Boeing. All rights reserved.

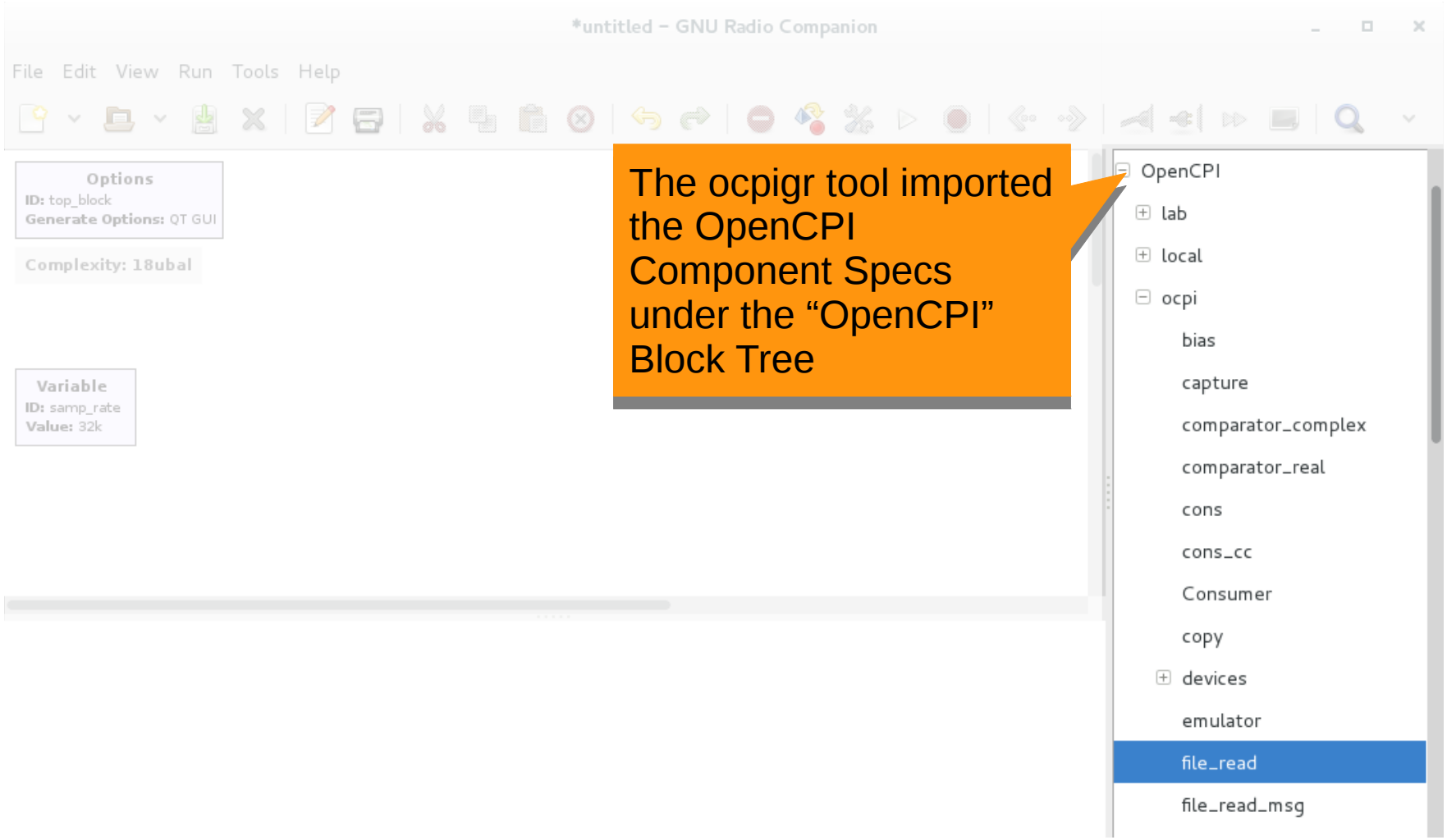
OpenCPI and GRC – ocpigr Tool

- Tool in the OpenCPI repo that will parse all OpenCPI workers/components found in the OCPI_LIBRARY_PATH and translate their XML into GRC block xml
- Creates an OpenCPI Container Platform block for each platform that was encountered during parsing
- Takes advantage of the GRC Domain concept by making a domain for each Platform encountered during parsing



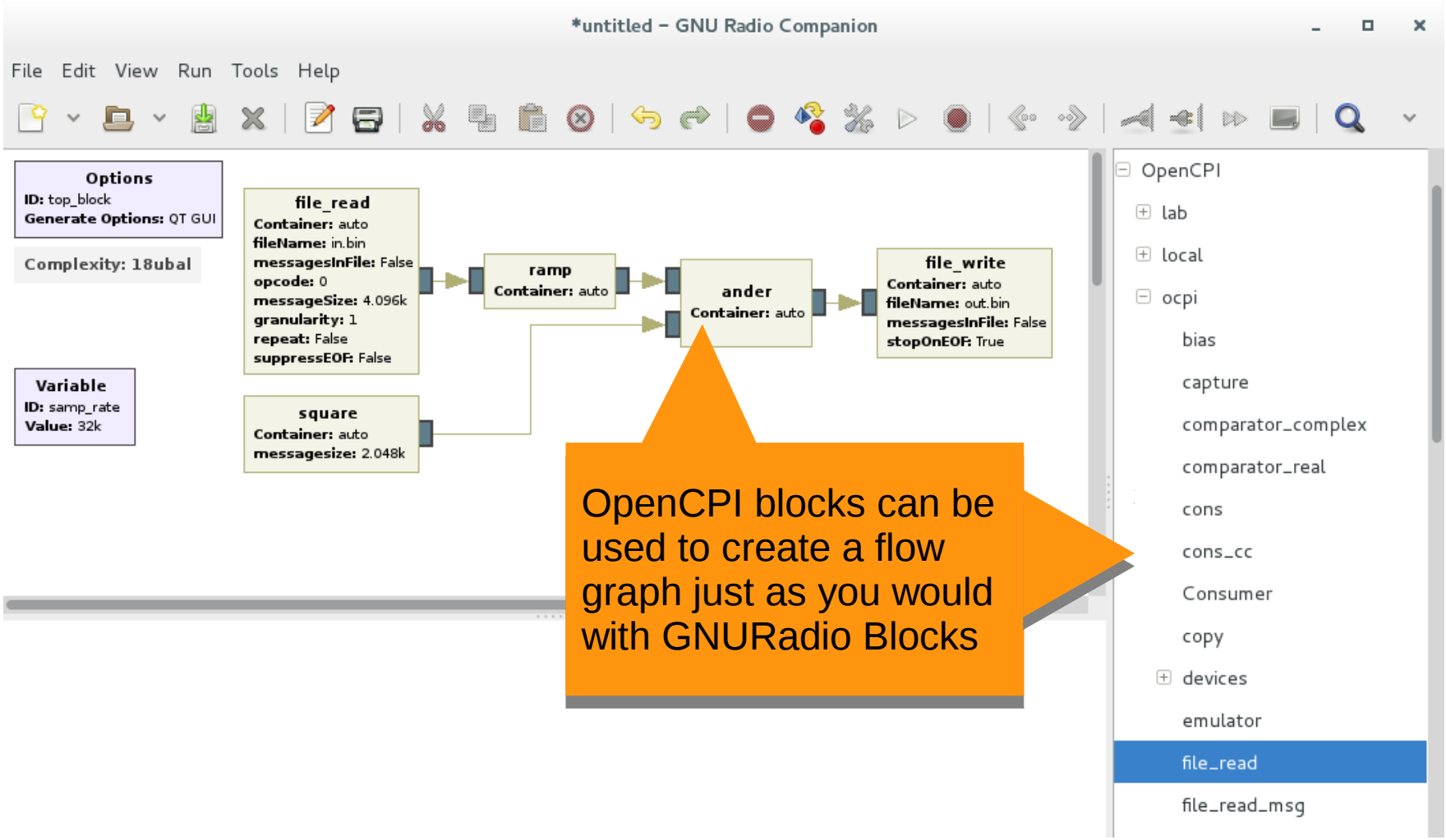
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – ocpigr Tool



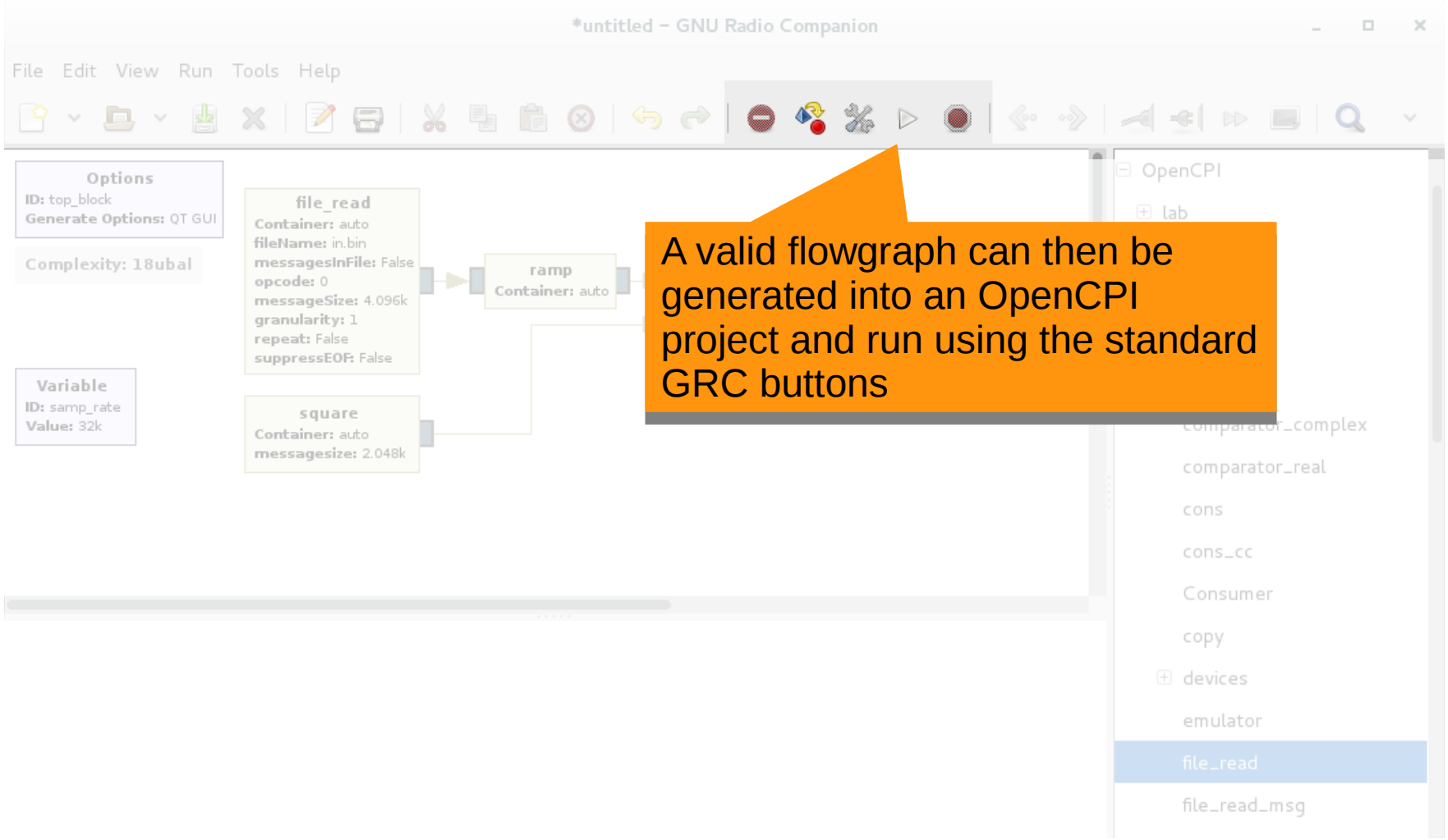
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Creating a Flowgraph



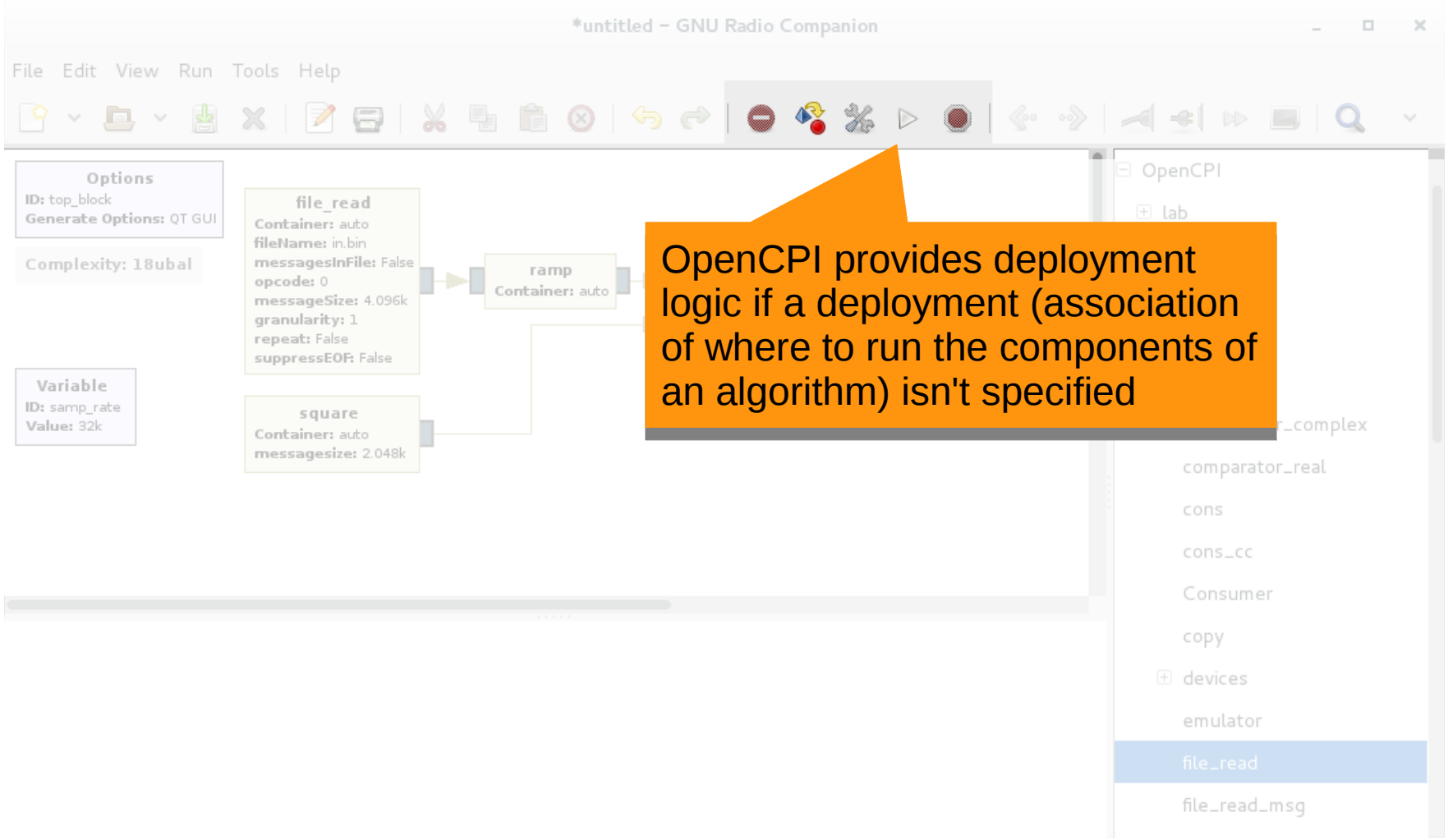
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Running a Flowgraph



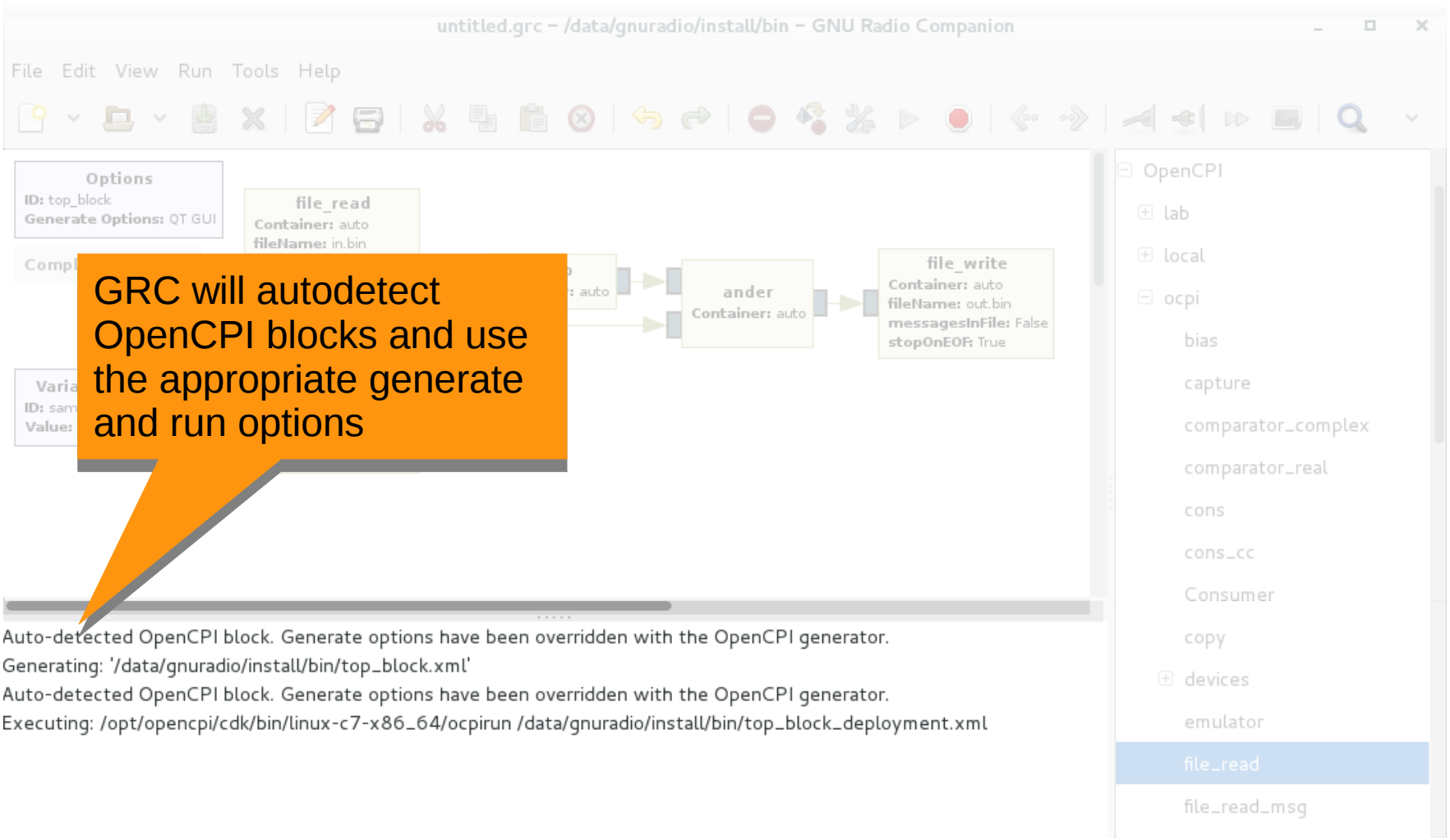
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Running a Flowgraph



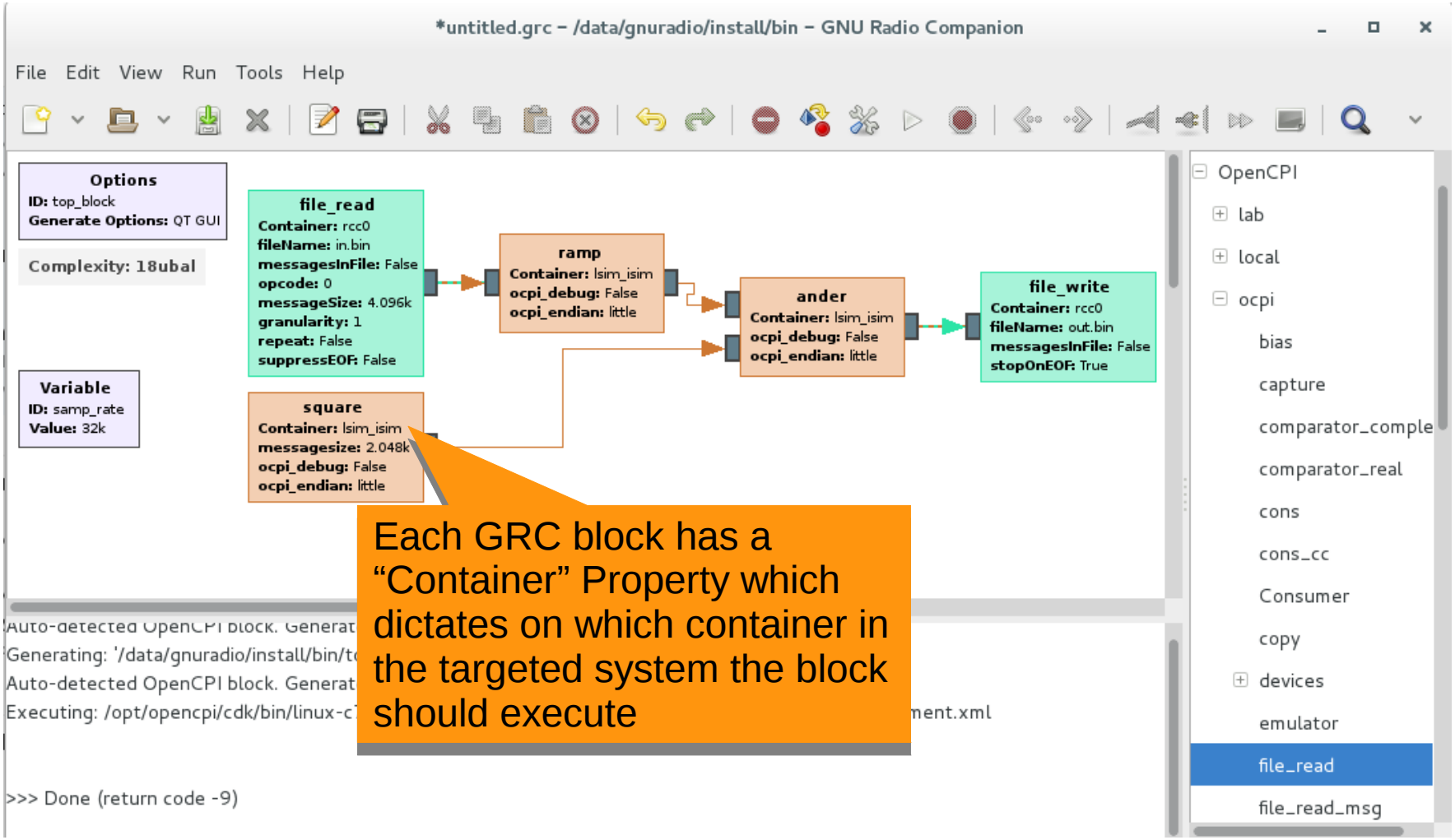
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Running a Flowgraph



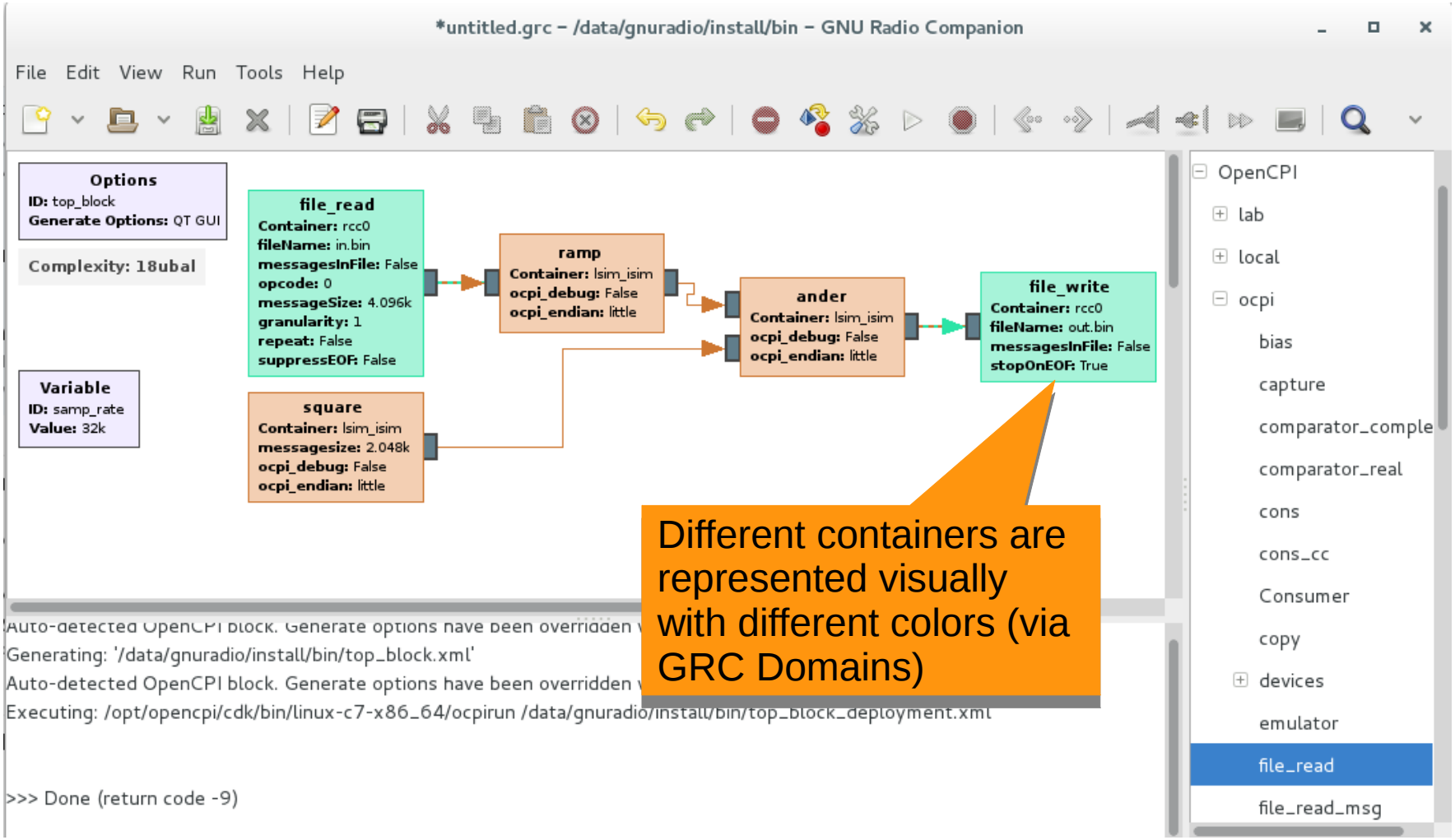
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Set Deployment



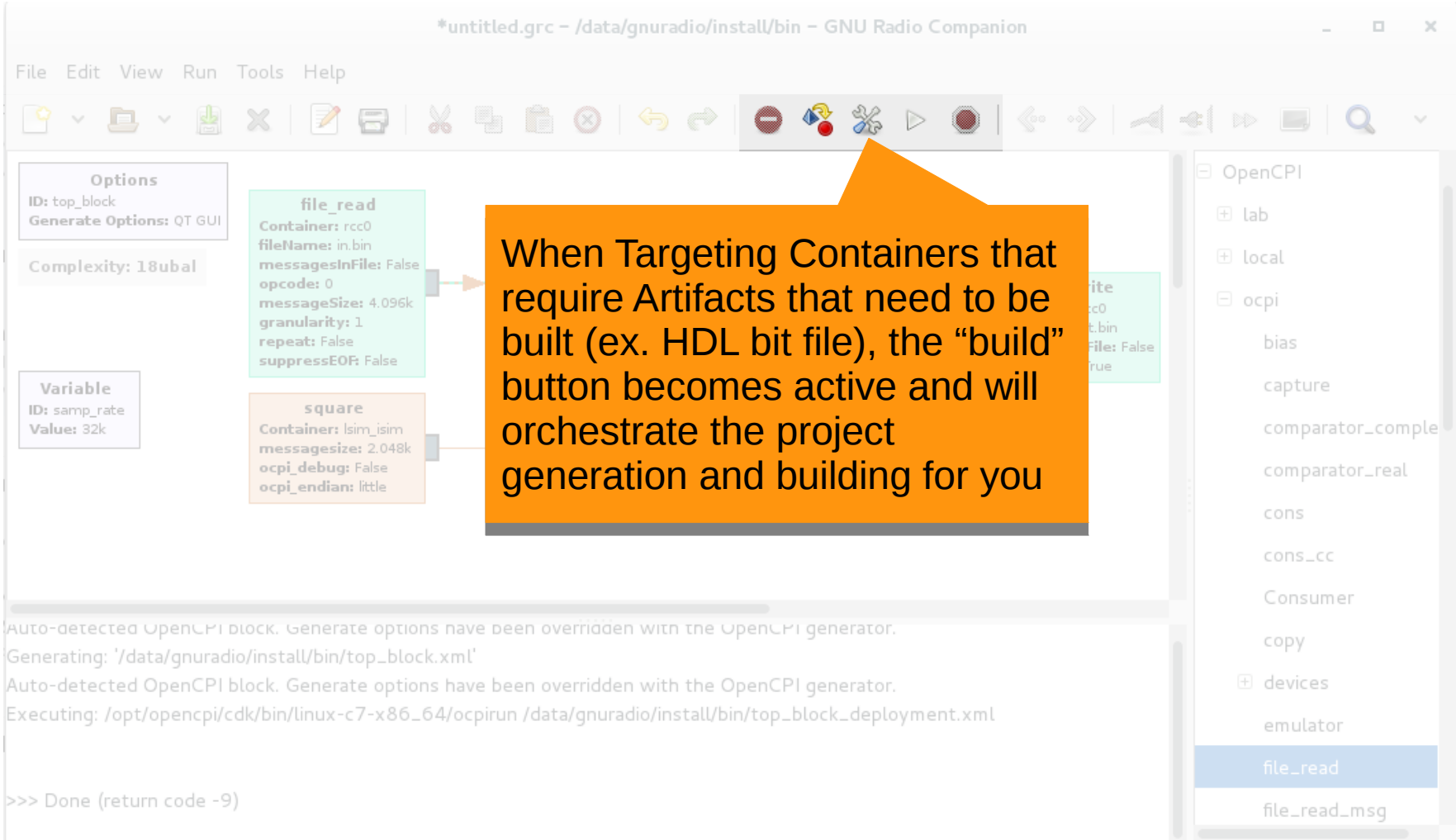
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Set Deployment



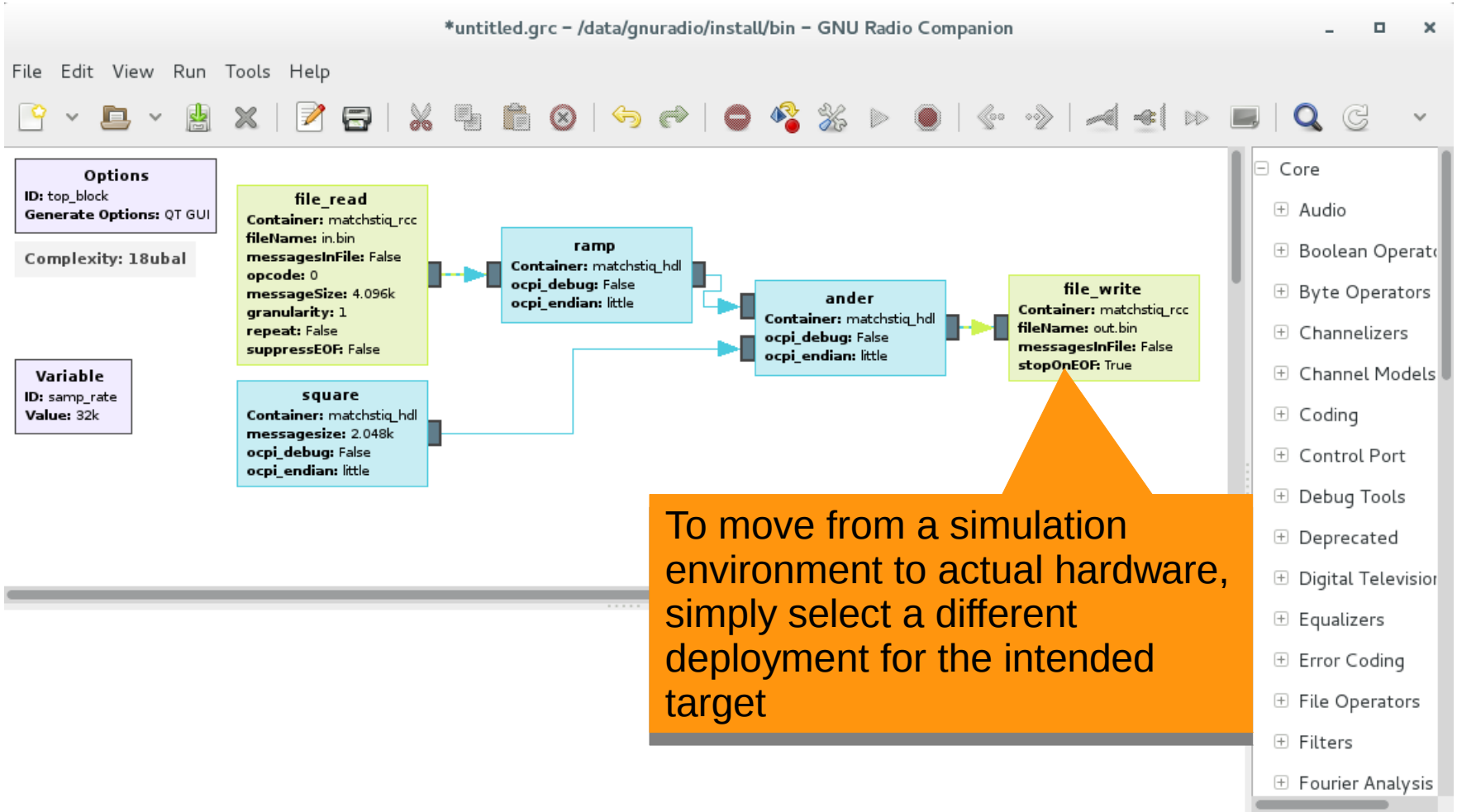
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Build Assembly



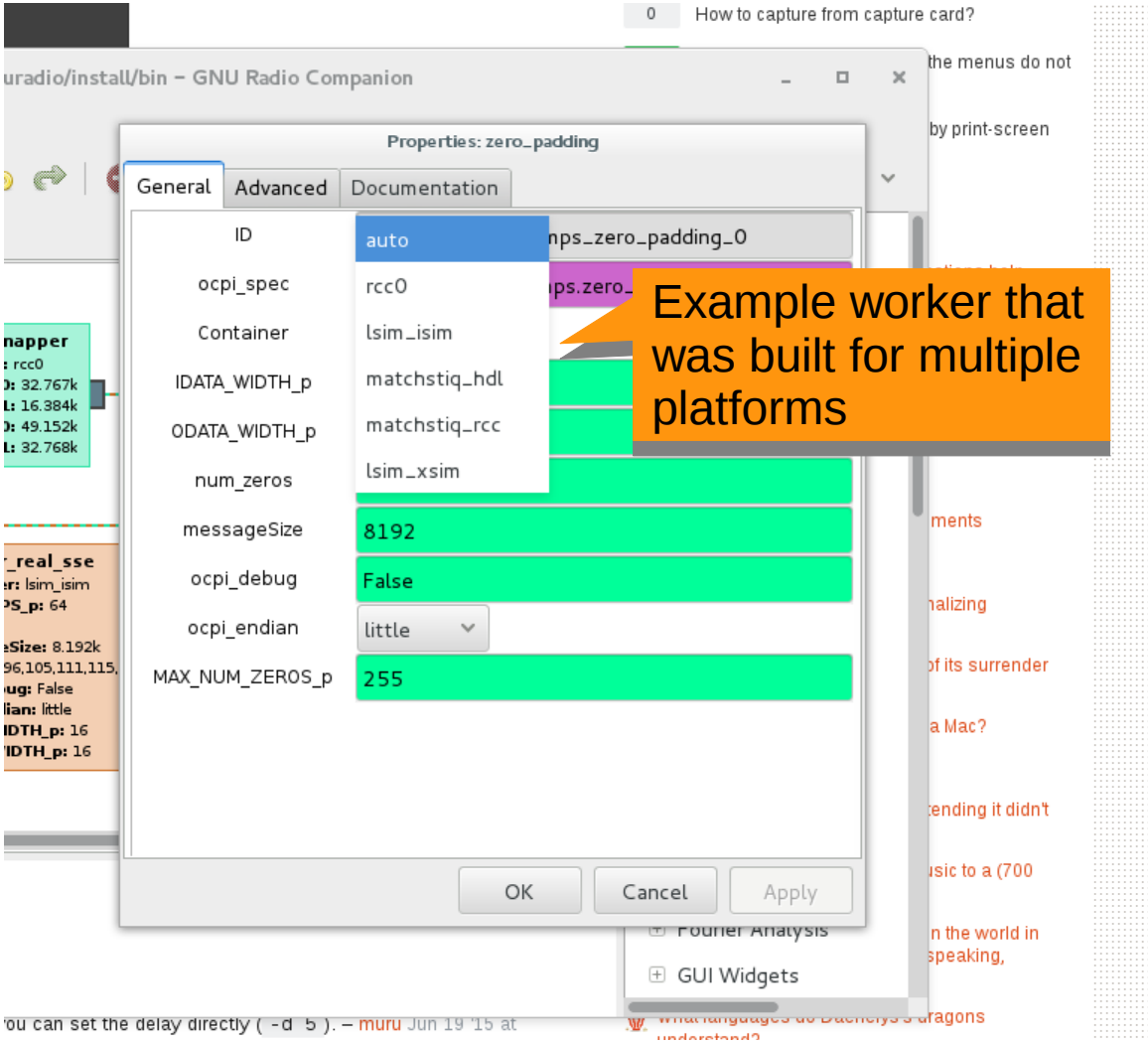
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Set Deployment



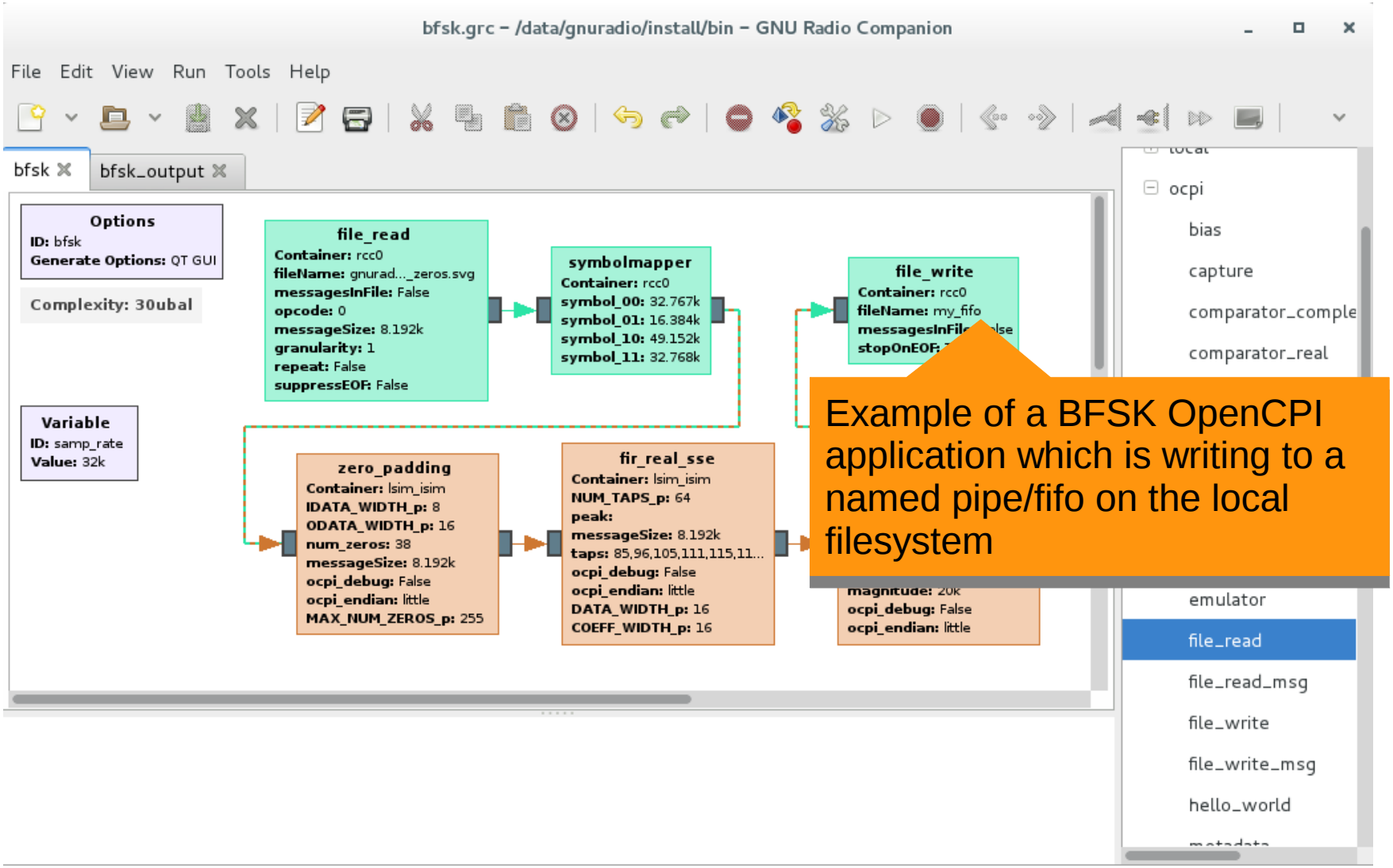
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Set Deployment



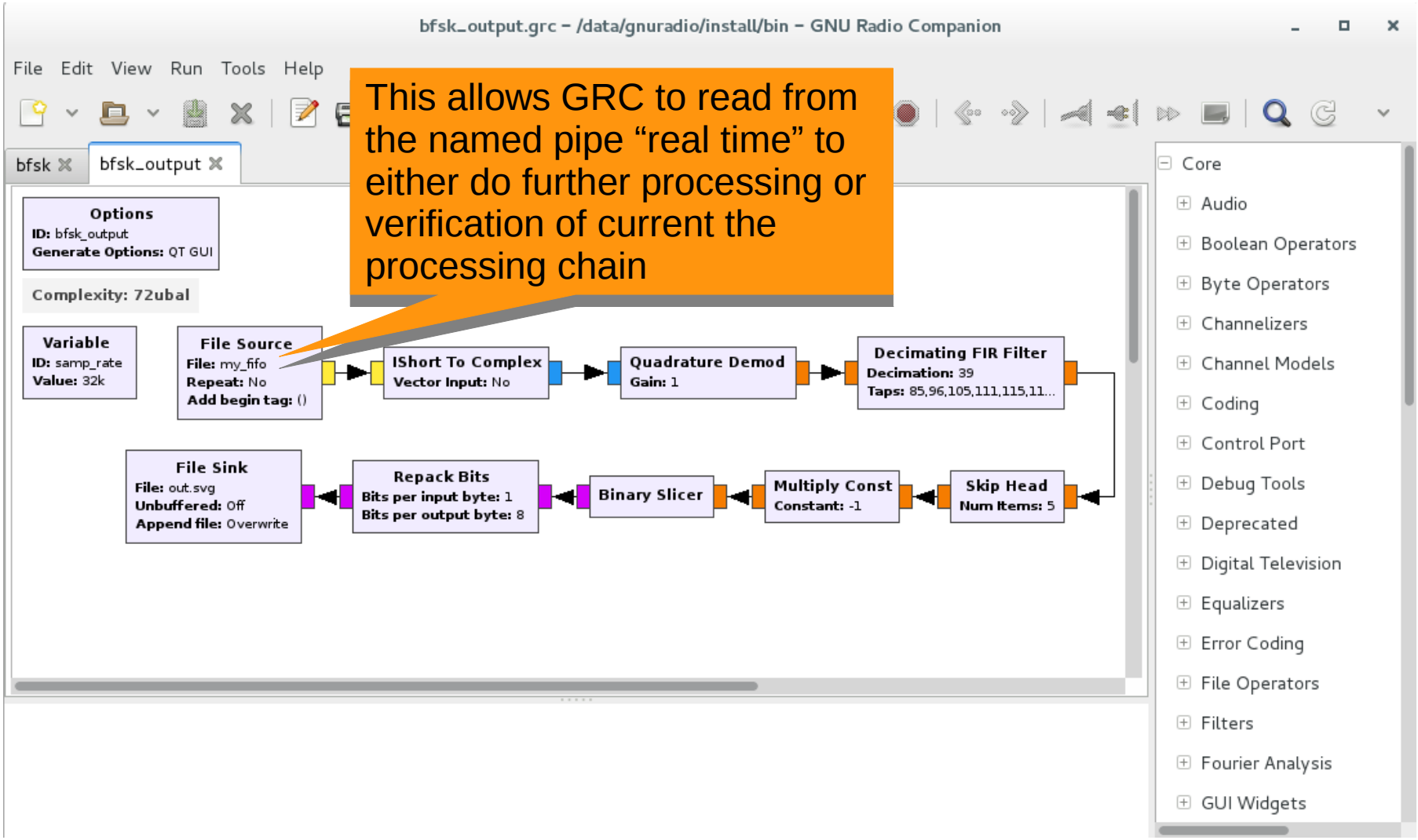
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Example BFSK App



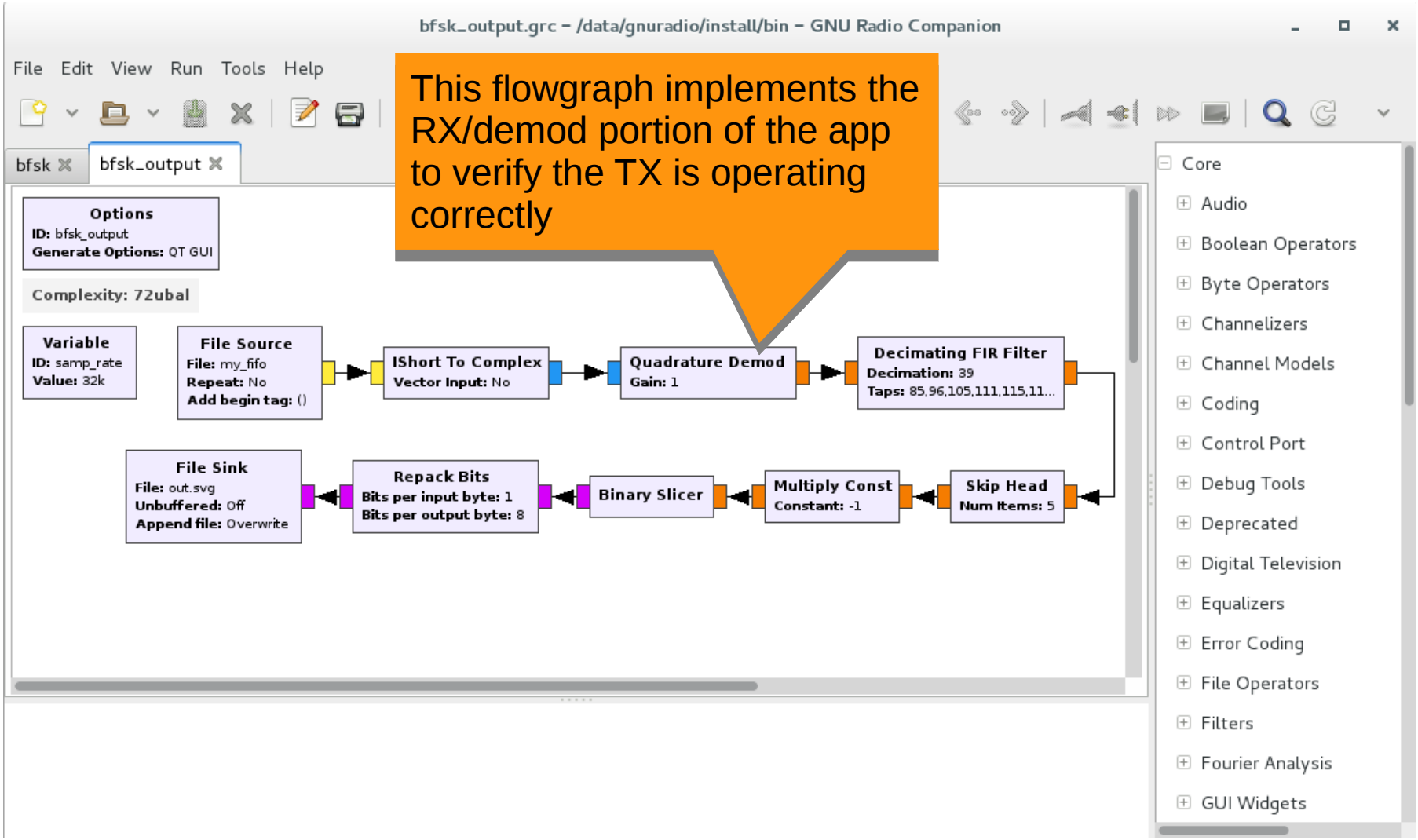
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Example BFSK App



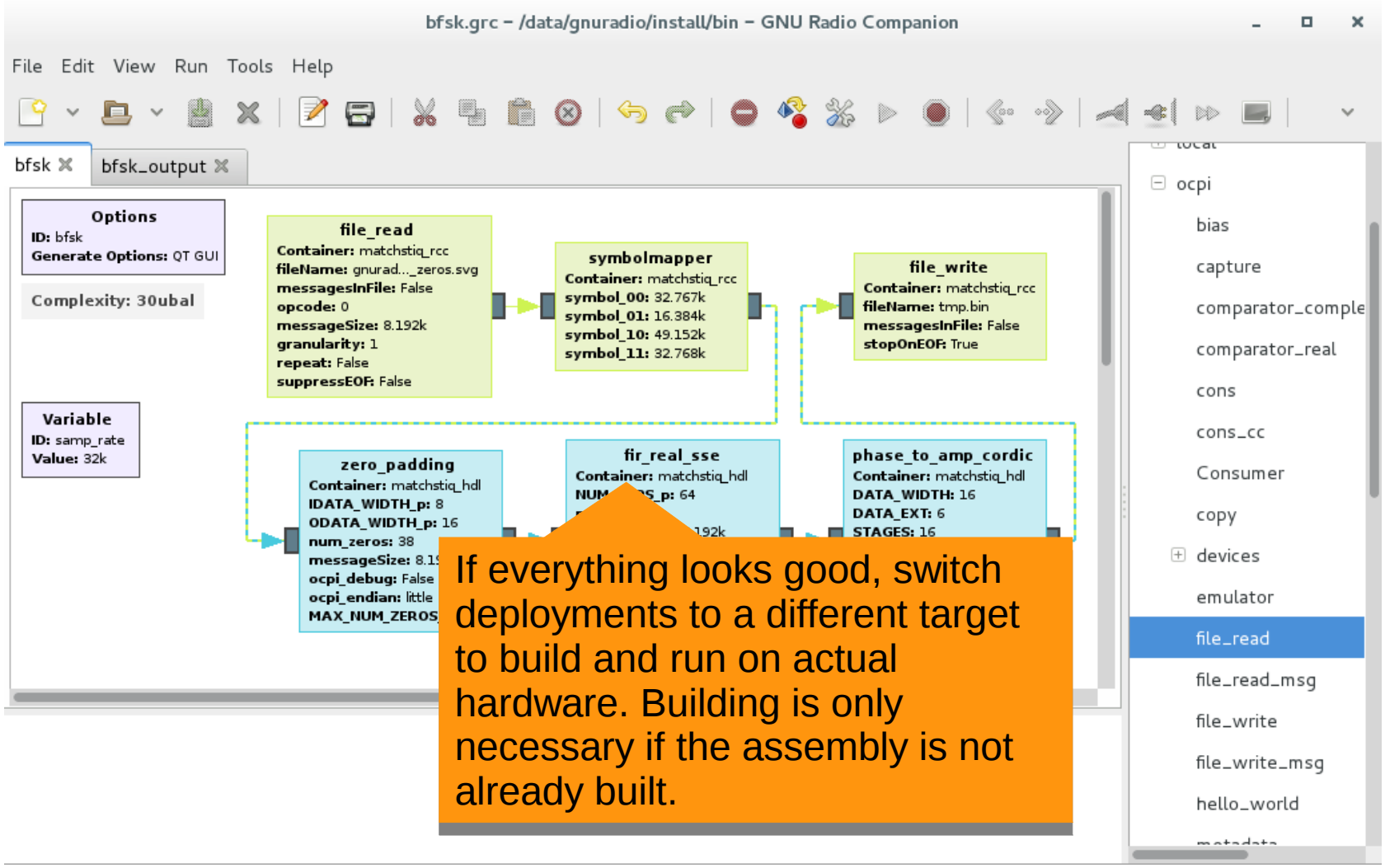
Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Example BFSK App



Copyright © 2017 Boeing. All rights reserved.

OpenCPI and GRC – Example BFSK App



Copyright © 2017 Boeing. All rights reserved.

Road Map for Future Development

- Have GRC and OpenCPI blocks talk directly or through a translation block
- Enable GRC to determine if the required artifacts are available and automatically build them if they are not
- Expose more features of OpenCPI to the GRC user
  - Select specific worker
  - Customize HDL Container
- Allow deployment across multiple systems on a network
- OpenCL/GPU support
- Support more platforms “out of the box”

Copyright © 2017 Boeing. All rights reserved.

How to Get More Information

- OpenCPI
  - https://www.opencpi.org
  - https://github.com/opencpi/opencpi (branch 2017.Q2)
  - issues@opencpi.org

Copyright © 2017 Boeing. All rights reserved.

Copyright © 2017 Boeing. All rights reserved.

Backup

Copyright © 2017 Boeing. All rights reserved.