

# Embedded System Deployment and Management

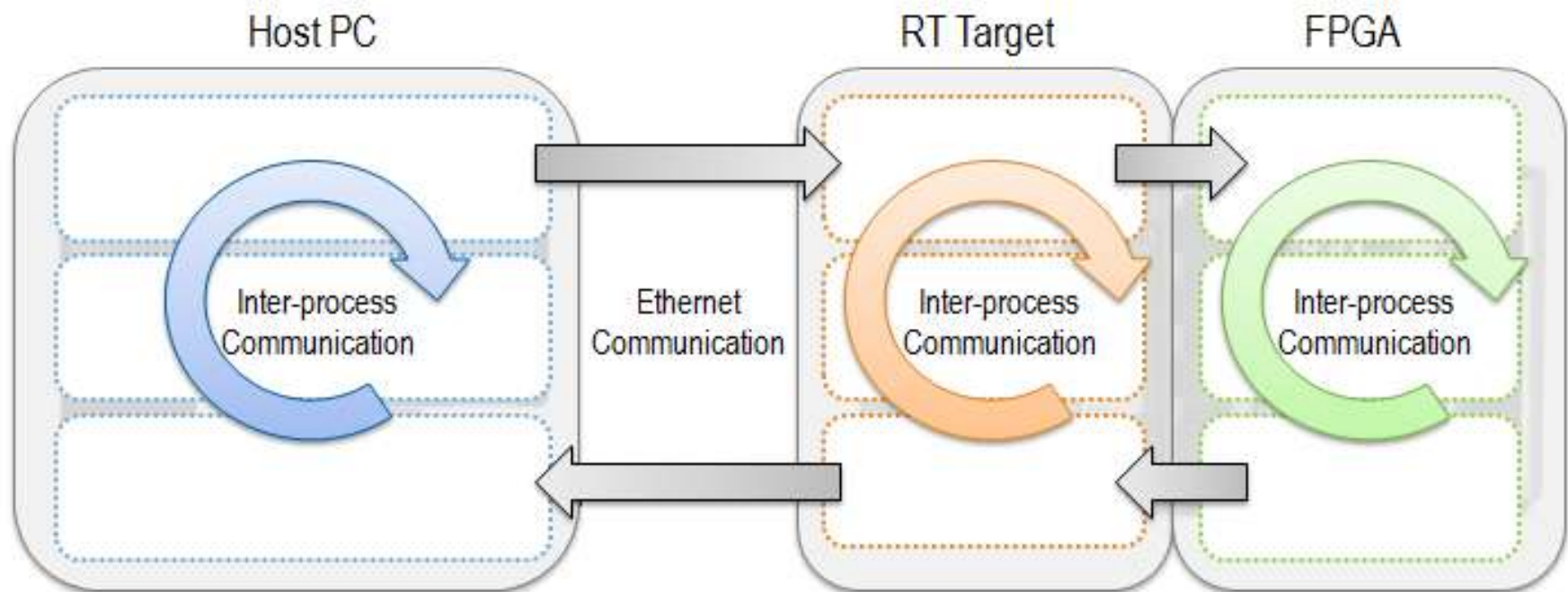
Richard Wasell

Applications Engineer, NI Norway

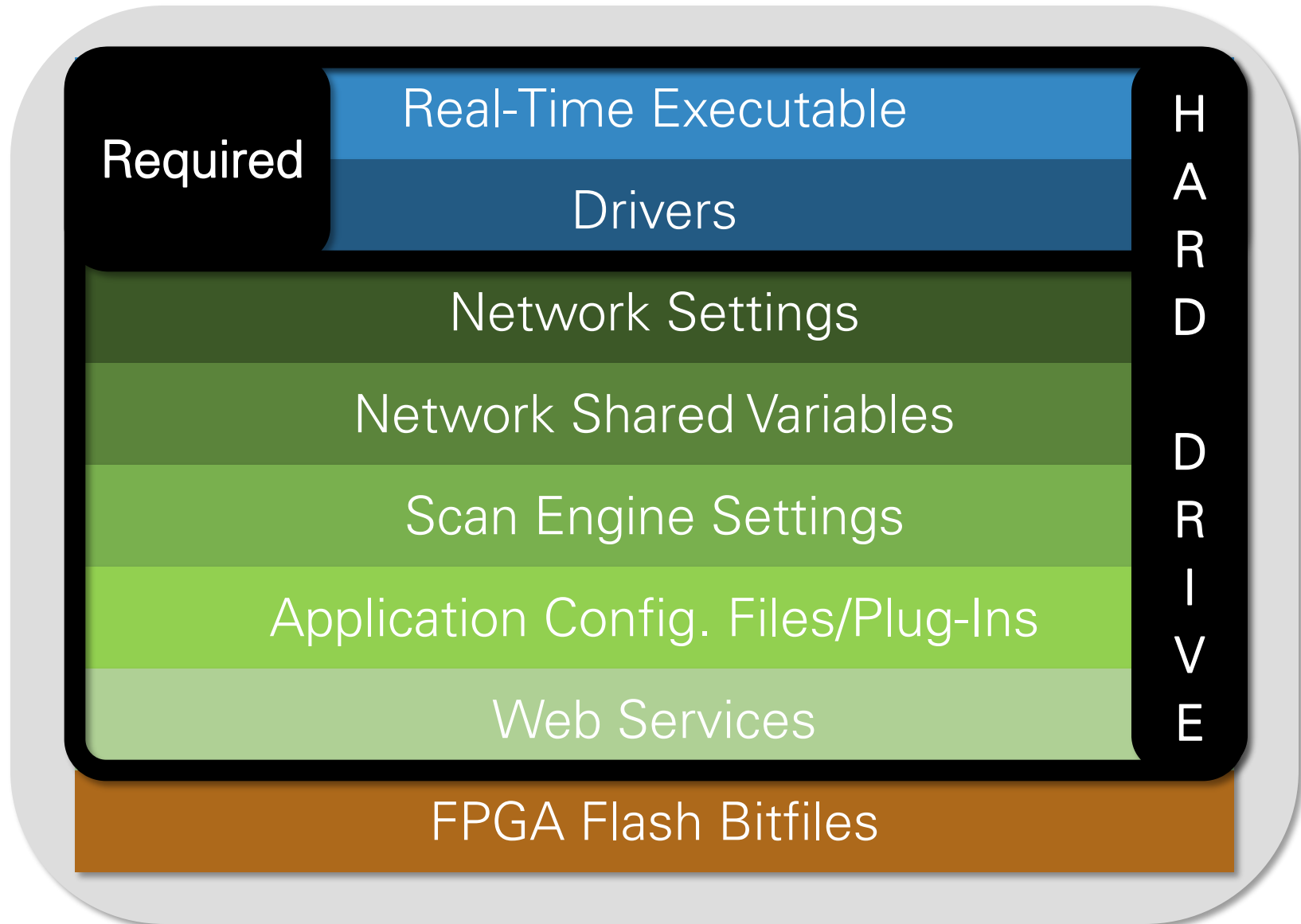
# Agenda

- Project Based Deployment Overview
  - FPGA Deployment
- Image vs. Component Based Updates
- Push Model
  - MAX & LabVIEW
  - System Images
  - Replication and Deployment Utility
  - Component Based Deployment
- Pull Model
  - USB Drive Image Deployment
  - Simple Pull Deployment Server
  - cRIO-9068 Deployment
- Conclusion

# RIO Architecture

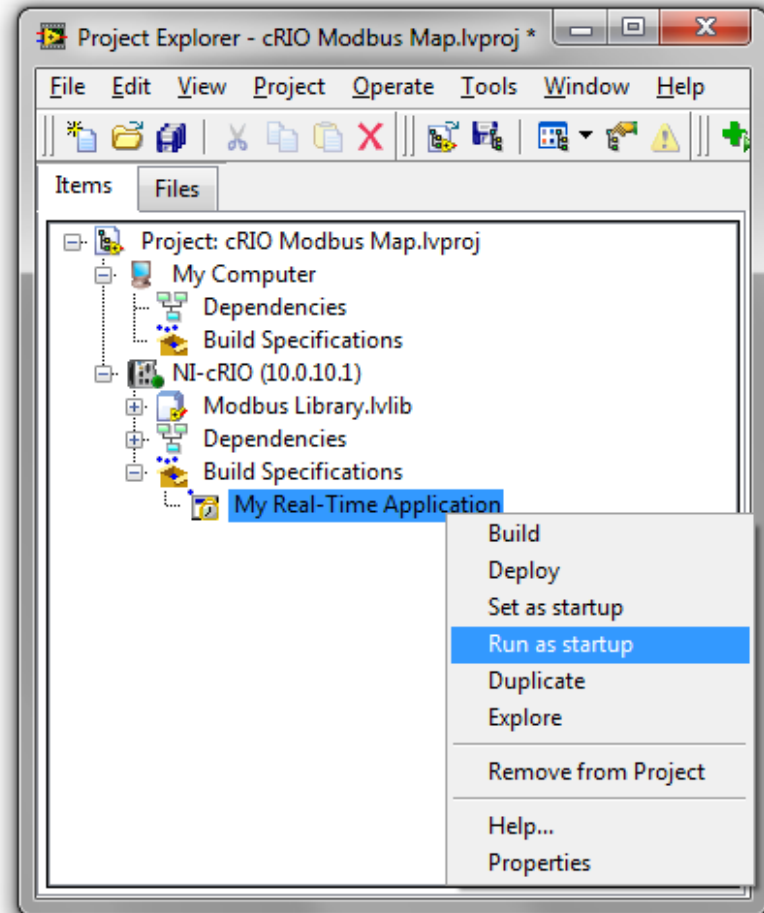


# RT System Software Components

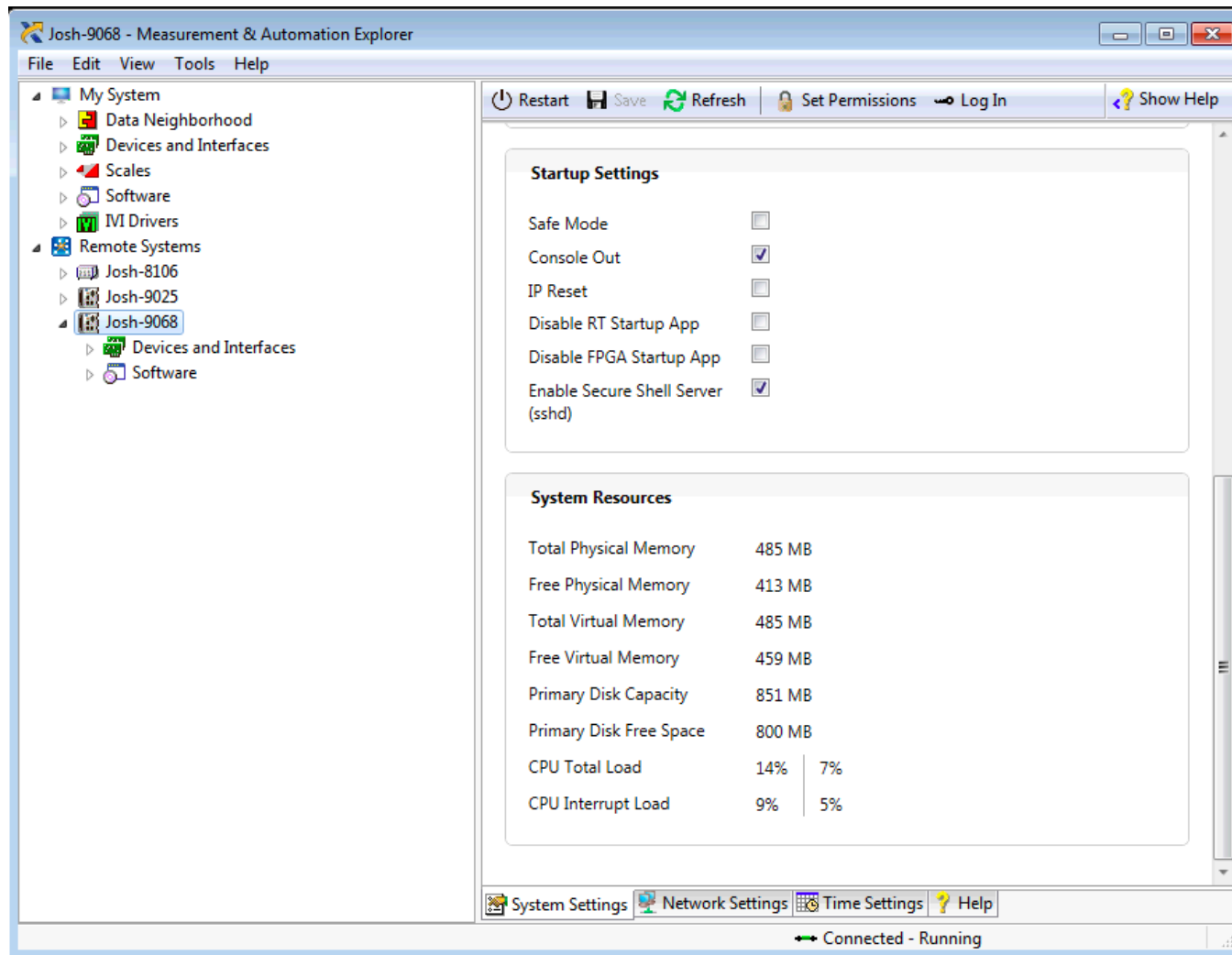


# Typical Deployment Using MAX and LabVIEW

1. Discover the target in MAX
2. Apply network settings
3. Install all necessary drivers
4. Change project IP Address
5. Deploy the controller settings
6. Deploy the application

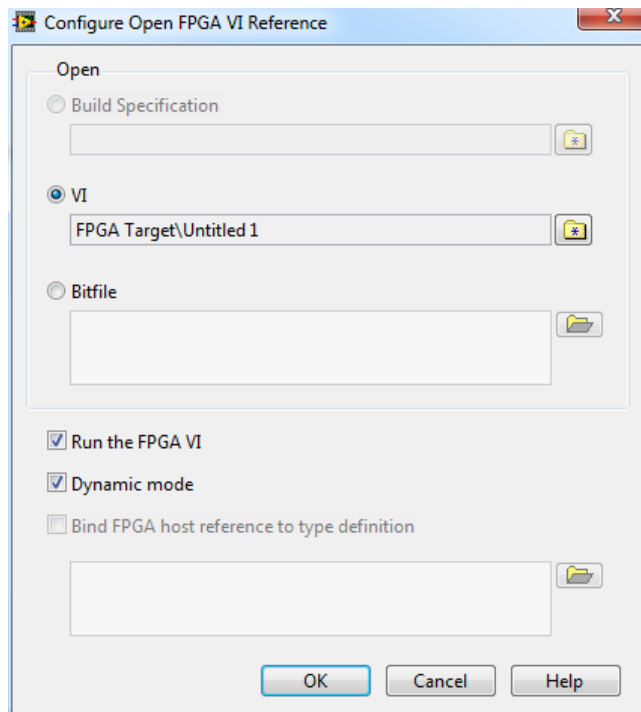


# New: Monitor Critical Controller Attributes

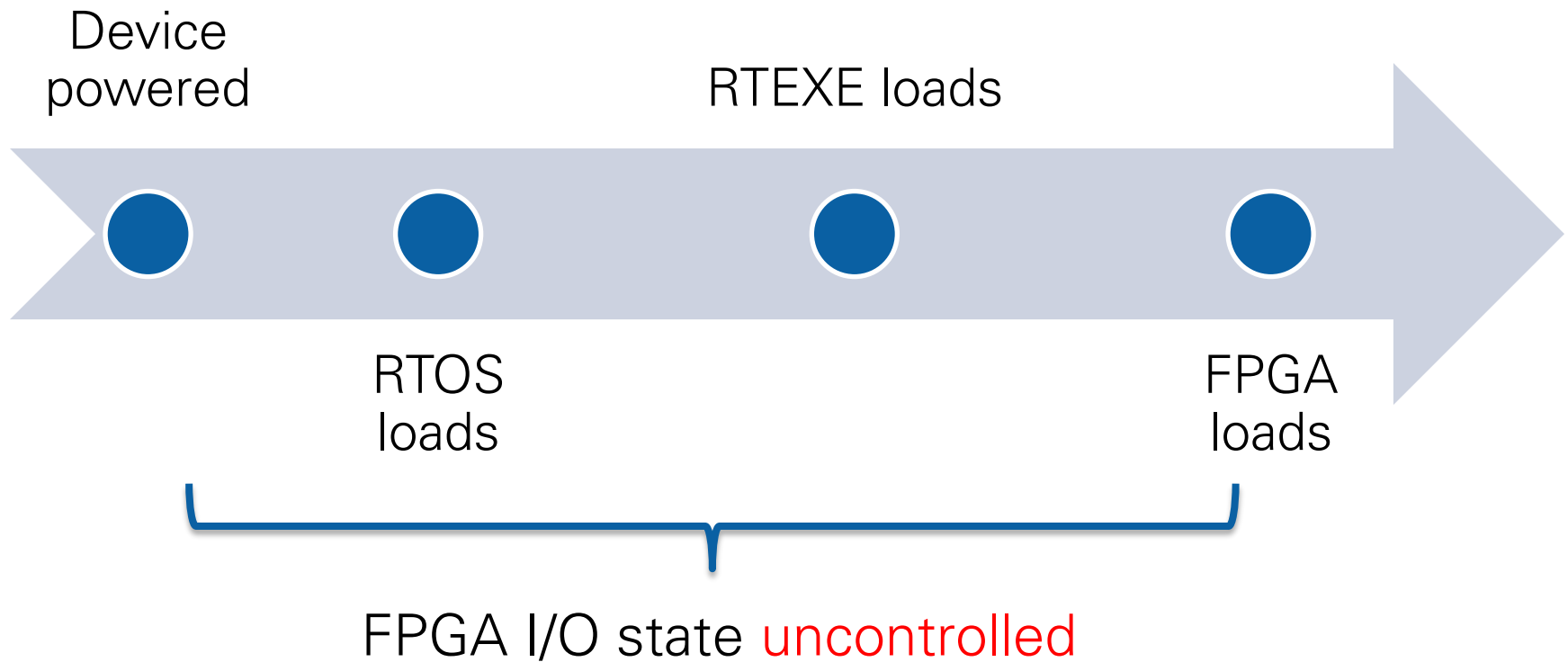


# LabVIEW Project Based Deployment

- What happens to FPGA code when deploying RTEXEs?



# Open FPGA VI Reference Timing

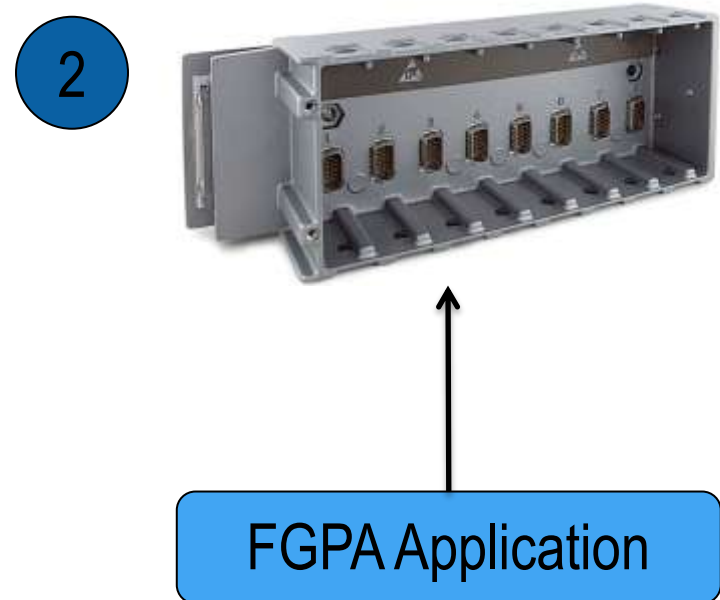
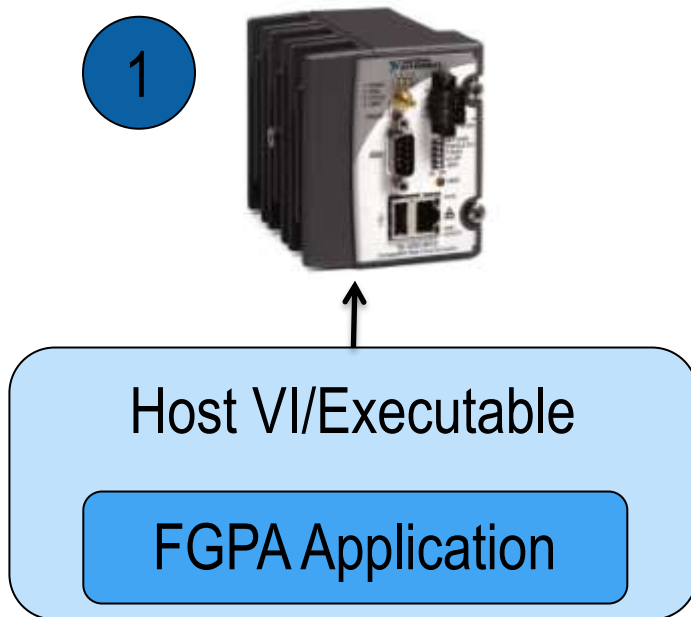




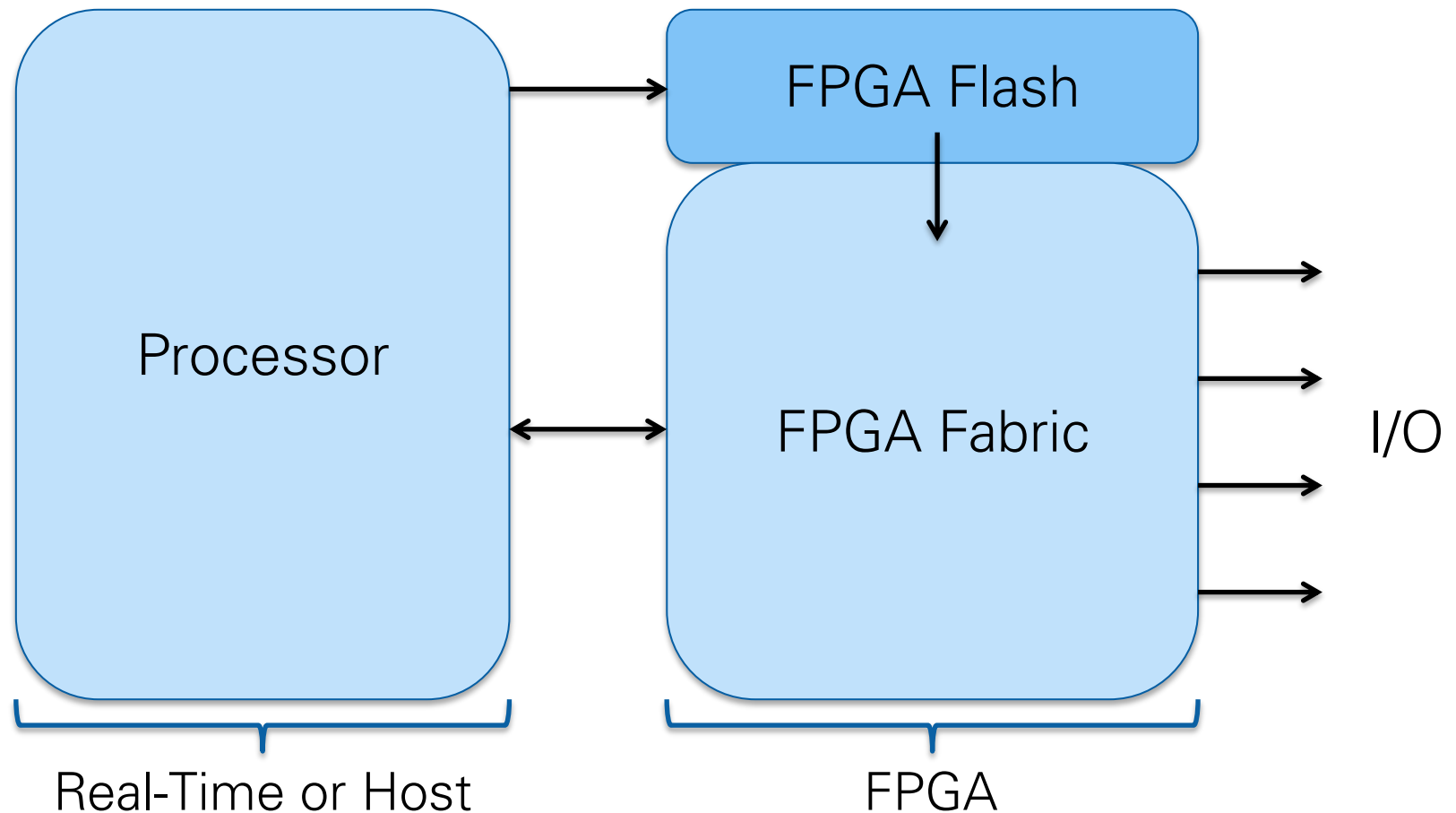
# FPGA Deployment

Two methods to deploy an FPGA bitfile:

1. Host can load bitfile with 'Open FPGA VI Reference.vi'
2. Load independently into FPGA flash memory

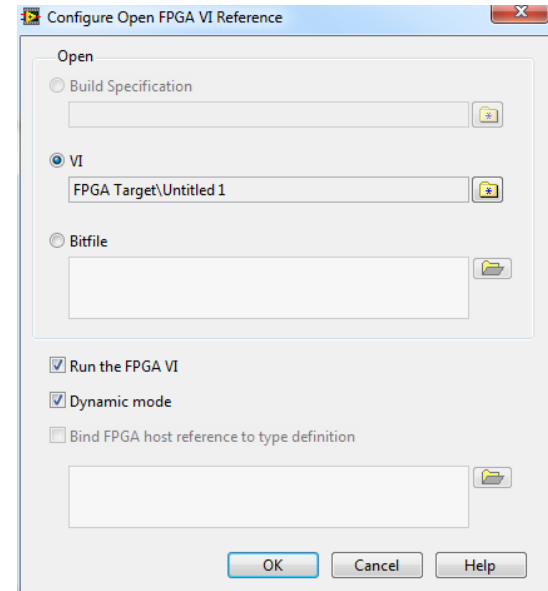


# RIO Hardware Architecture Overview



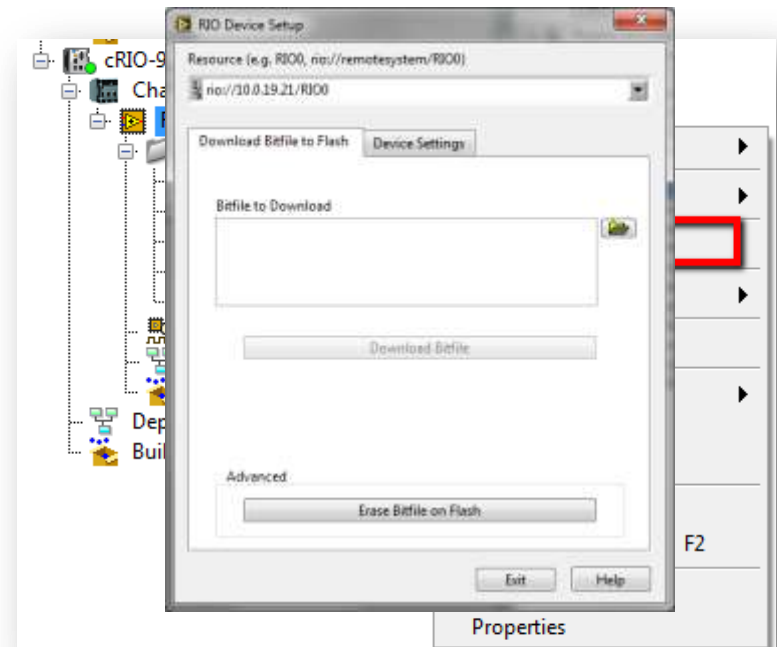
# FPGA Deployment: Host Reference

- Bitfile is saved in host application with 'Open FPGA VI Reference.vi'
- In execution, Open FPGA function checks if bitfile is loaded and loads if not already loaded
- Caveat: bitfile isn't loaded and run until OS and application load



# FPGA Deployment: FPGA Flash

- Alternative is to deploy FPGA bitfile to flash
- Right Click the FPGA Target in the project and select **RIO Device Setup...**
- Programmatically, use [NI-RIO System Replication VIs](#) found on NI Developer Zone (Title: FPGA Target System Replication)



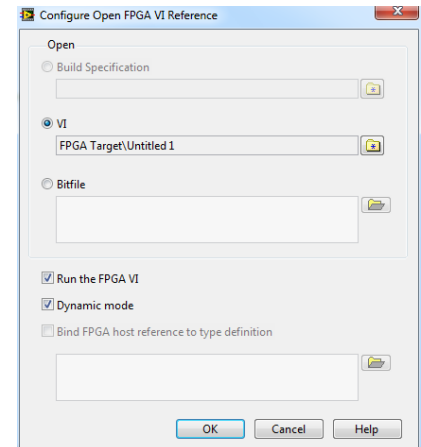
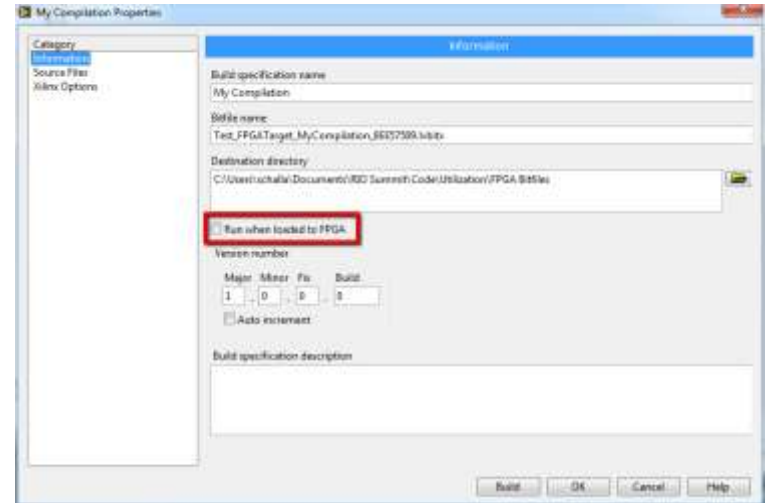
Set RIO Device Settings.vi



Download Bitfile.vi

# FPGA Deployment

- Bitfile can run on device power up and/or reboot:
  - **Note:** Must compile with 'Run when loaded to FPGA' checked (not default)
- RTEXE can still connect to FGPA bitfile



# New: Manage FPGA Bit Files on NI Linux Real-Time

- Update and erase the FPGA bit files on NI Linux Real-Time targets programmatically, from MAX, and the web

The image displays two overlapping software windows from National Instruments. The background window is 'NI cRIO-9068 "RIO0" - Measurement & Automation Explorer', showing a tree view of system components. The foreground window is 'Josh-9068 : NI Web-based Configuration & Monitoring', showing a web interface for system configuration.

**Hardware Configuration (MAX Tree View):**

- My System
  - Data Neighborhood
  - Devices and Interfaces
  - Scales
  - Software
  - IVI Drivers
  - Remote Systems
    - Josh-8106
    - Josh-9025
    - Josh-9068
      - Devices and Interfaces
        - NI cRIO-9068 "RIO0"
          - ASRL1::INSTR
          - ASRL2::INSTR
          - ASRL3::INSTR
        - Software

**Josh-9068 : System Configuration (Web Interface):**

Search: [ ] Save Refresh

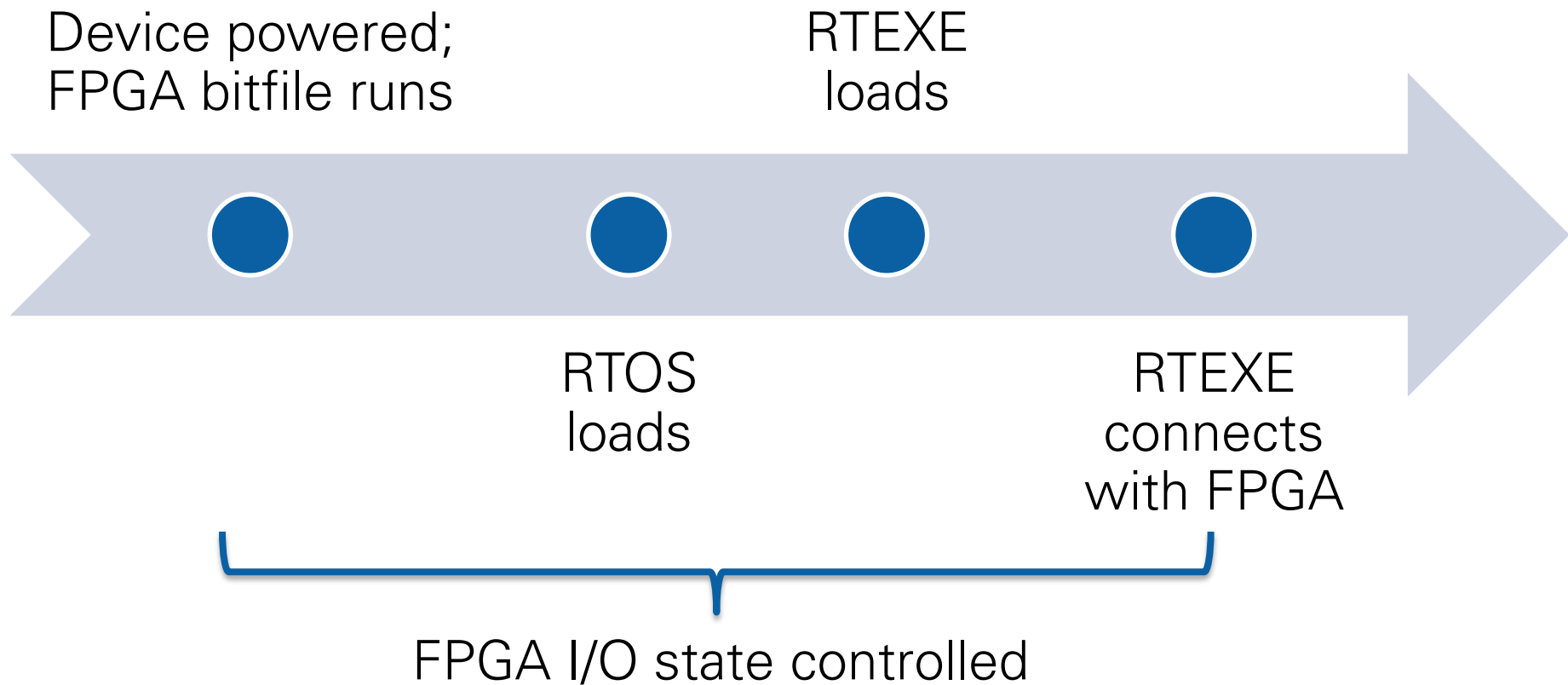
Icon	Device Name	Internal Name
Computer	cRIO-9068	Name: Josh-9068
Module	NI cRIO-9068	Name: RIO0
Module	ASRL1::INSTR	Name: ASRL1::INSTR
Module	ASRL2::INSTR	Name: ASRL2::INSTR
Module	ASRL3::INSTR	Name: ASRL3::INSTR

**Settings**

Property	Value
Name	RIO0
Vendor	National Instruments
Model	NI cRIO-9068
Serial Number	01856EAC
Status	Present

Erase Firmware Update Firmware

# FPGA Flash Deployment Timing



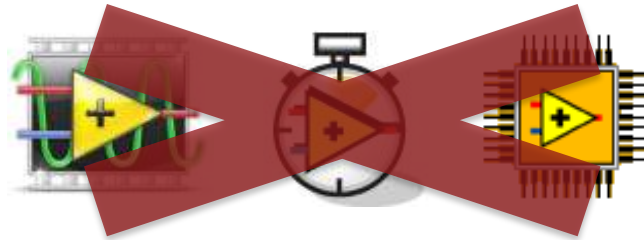
# Difficulties

What about project-based deployment in the context of:

1.



2.



3.



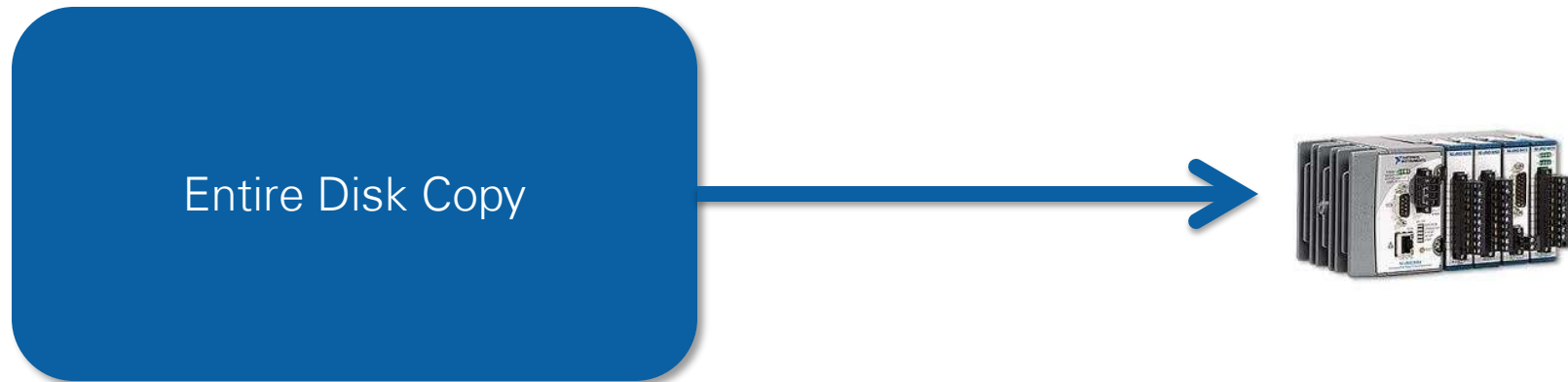
4.



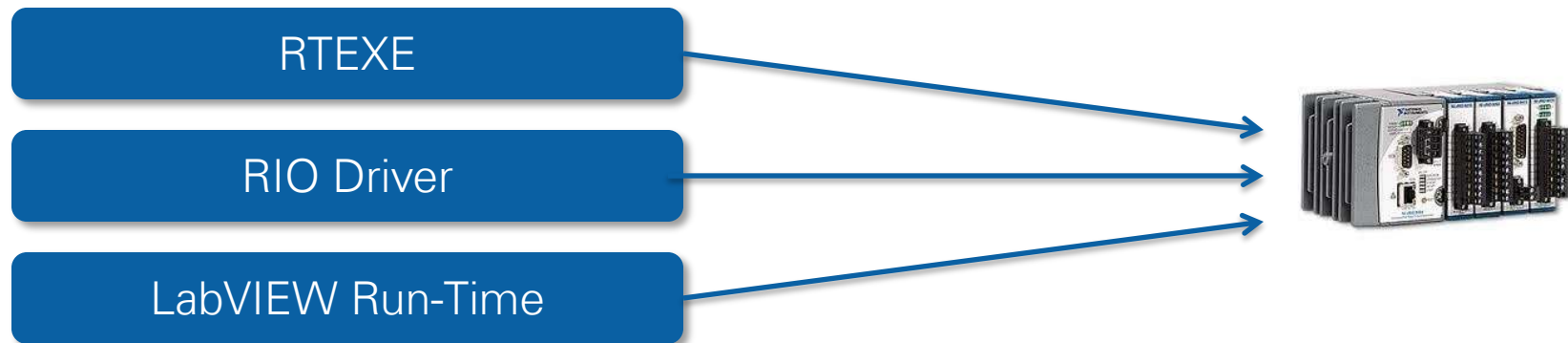


# Image vs. Component-based updates

## Image-Based Updates



## Component-Based Updates



# Image vs. Component-based Updates

## Image-Based Updates



Entire Disk Copy

### Simple

- All inclusive
- Only one deploy function
- Blacklisting for customization
- Ideal for factory installations

## Component-Based Updates



RTEXE



RIO Driver



LabVIEW Run-Time

### Flexible

- Update exactly what you want
- Efficient network bandwidth
- Controller agnostic
- More powerful tool for updates

# Push Model

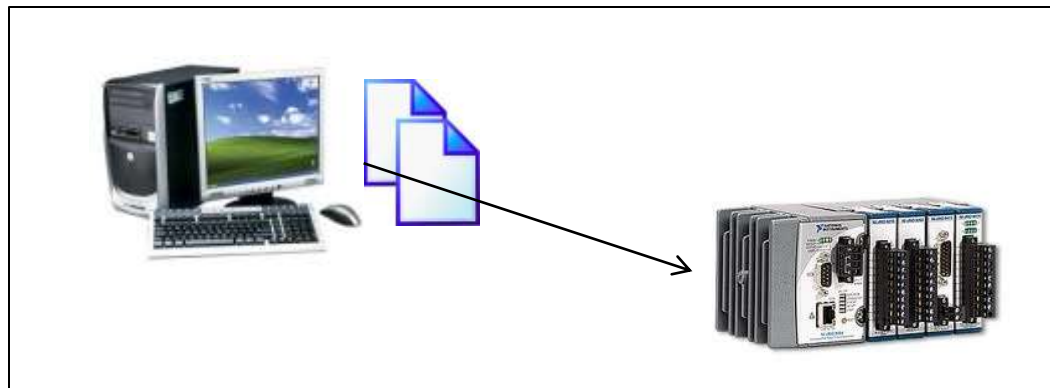
## Pros:

- Simple infrastructure
- Easy to develop
- Update code doesn't reside on the controller
- **Approach of nearly all NI APIs and RAD**



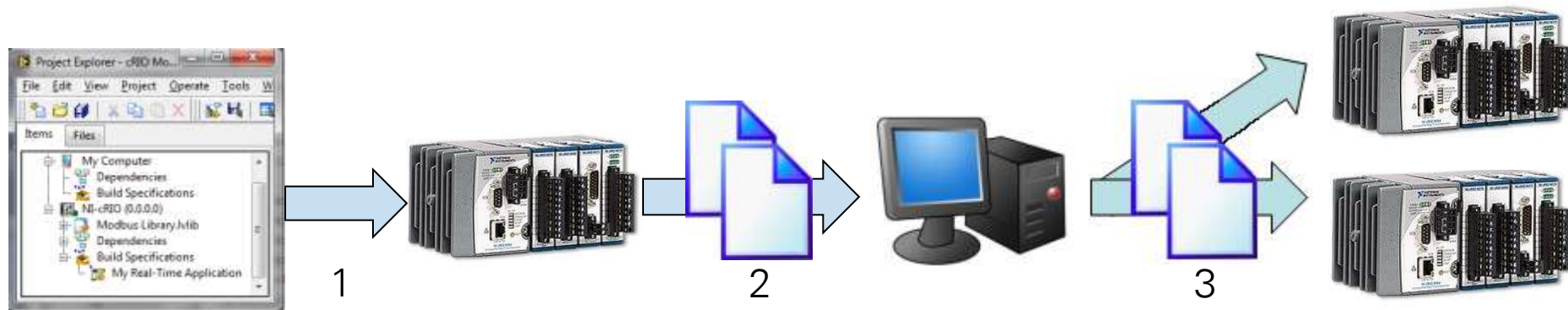
## Cons:

- Must know of or discover all targets
- Possible to perform updates at a bad time

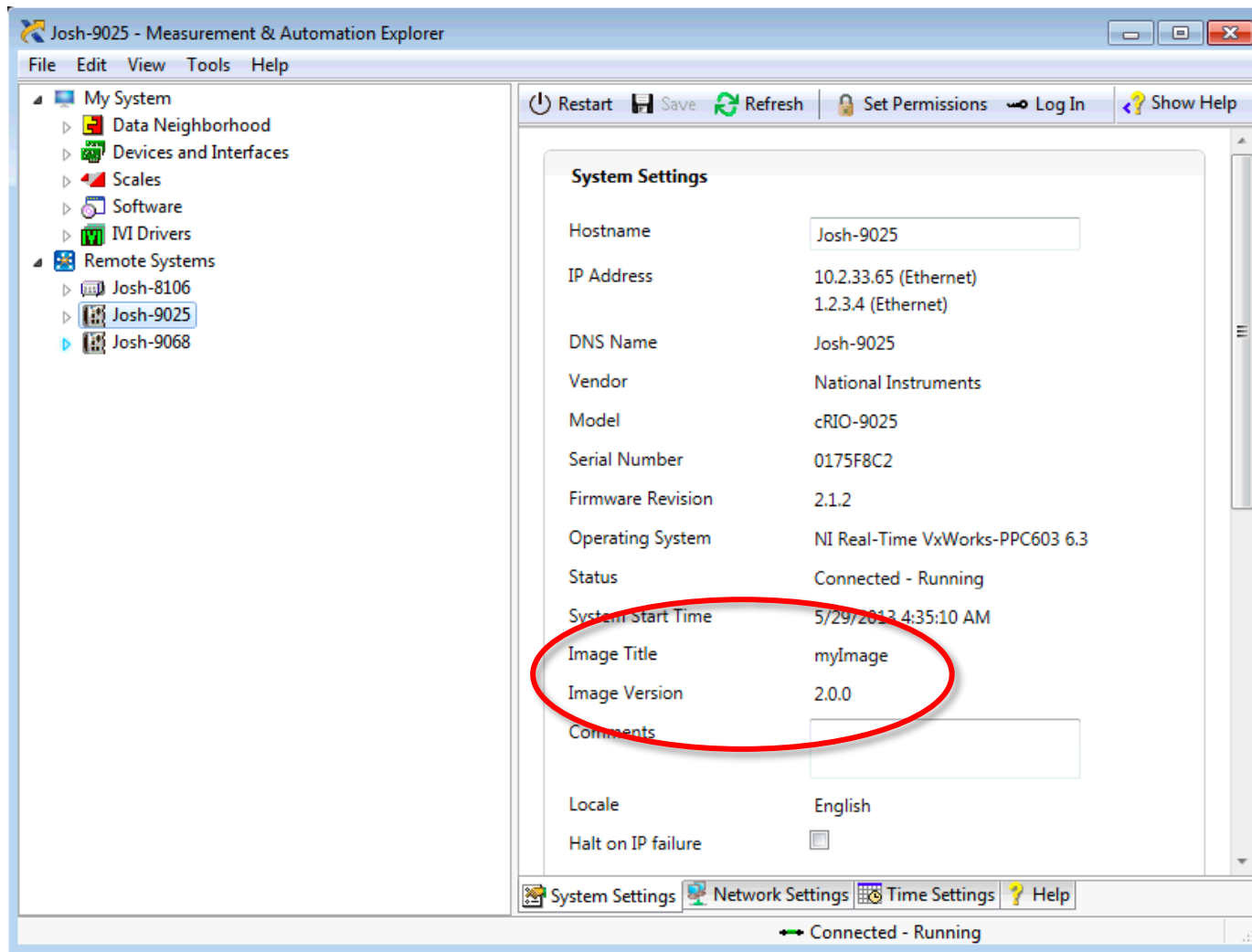


# Imaging

1. Deploy built application to a controller from project
2. Create Disk Image from this controller
  - Image packages every file on the controller hard drive in a zip file saved on the host machine
3. Deploy that image to one or more targets



# New: View Image Meta-Data from MAX or Web Interface



# Replication and Deployment Utility (RAD)

- Open source **image-based** deployment utility
- Use 'as-is' or as a starting point for a custom solution
- Developer Zone: [Replication and Deployment \(RAD\) Utility](#)

The screenshot displays the Replication and Deployment Utility (RAD) application window. The interface is divided into two main sections: Deployment Targets and Application Images. The Deployment Targets section on the left contains a table with columns for Host Name, IP Address, Serial Number, and Model. The Application Images section on the right contains a table with columns for Application Name, Version, Date/Time, and Model. Both sections include 'Refresh' and 'Settings' buttons. Between the tables are 'Deploy', 'Retrieve', and 'Compare' buttons. At the bottom of each section are 'Add Target', 'Select All', and 'Configure' buttons. The Application Images table also has an 'Exit' button in the top right corner.

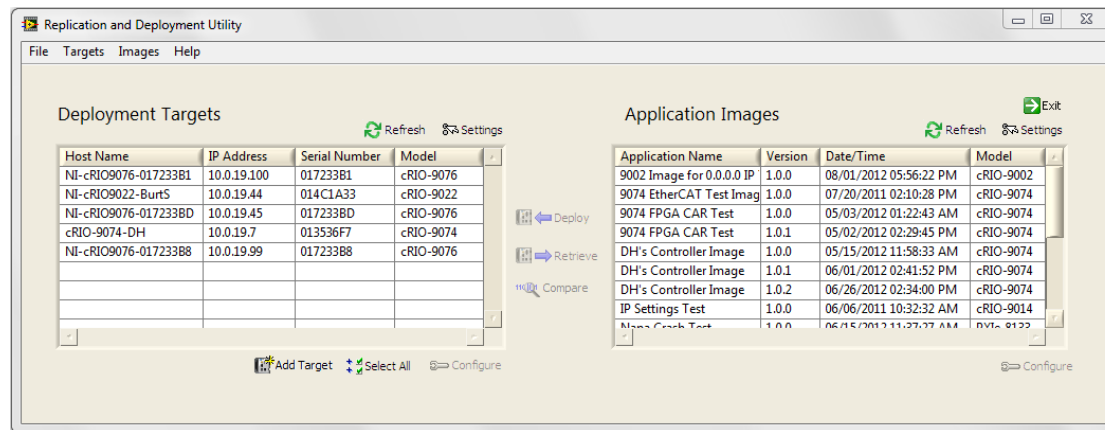
**Deployment Targets**

Host Name	IP Address	Serial Number	Model
NI-cRIO9076-017233B1	10.0.19.100	017233B1	cRIO-9076
NI-cRIO9022-BurtS	10.0.19.44	014C1A33	cRIO-9022
NI-cRIO9076-017233BD	10.0.19.45	017233BD	cRIO-9076
cRIO-9074-DH	10.0.19.7	013536F7	cRIO-9074
NI-cRIO9076-017233B8	10.0.19.99	017233B8	cRIO-9076

**Application Images**

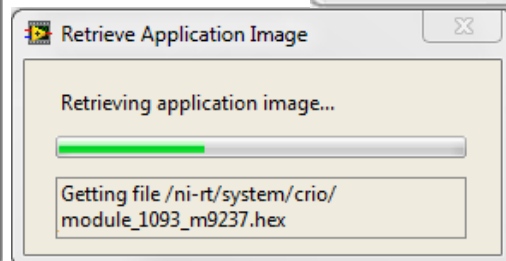
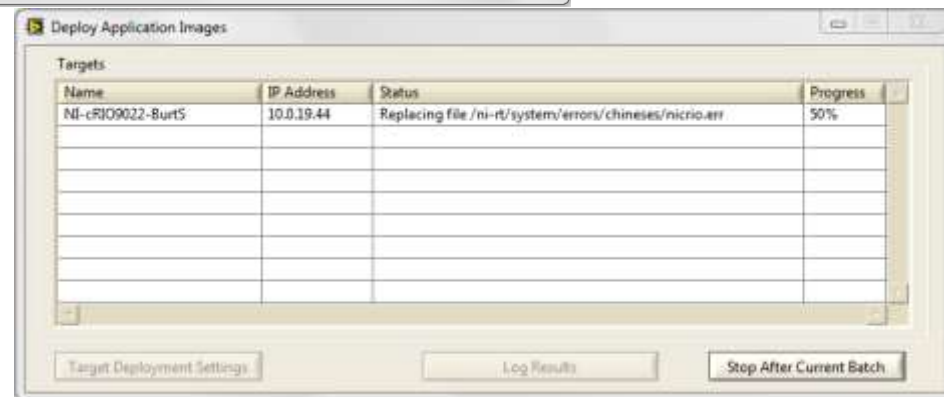
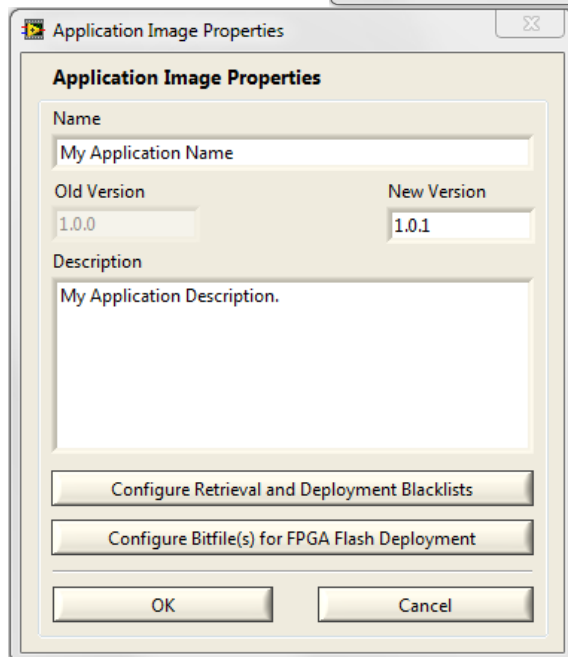
Application Name	Version	Date/Time	Model
9002 Image for 0.0.0.0 IP	1.0.0	08/01/2012 05:56:22 PM	cRIO-9002
9074 EtherCAT Test Image	1.0.0	07/20/2011 02:10:28 PM	cRIO-9074
9074 FPGA CAR Test	1.0.0	05/03/2012 01:22:43 AM	cRIO-9074
9074 FPGA CAR Test	1.0.1	05/02/2012 02:29:45 PM	cRIO-9074
DH's Controller Image	1.0.0	05/15/2012 11:58:33 AM	cRIO-9074
DH's Controller Image	1.0.1	06/01/2012 02:41:52 PM	cRIO-9074
DH's Controller Image	1.0.2	06/26/2012 02:34:00 PM	cRIO-9074
IP Settings Test	1.0.0	06/06/2011 10:32:32 AM	cRIO-9014
None Crash Test	1.0.0	06/15/2012 11:27:27 AM	DVI- 8123

# RAD: Image Management



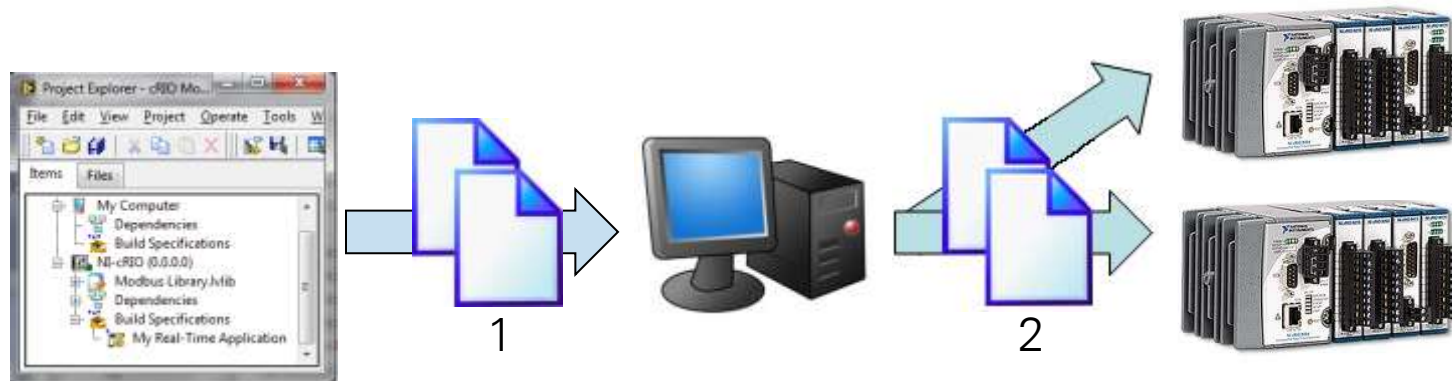
Retrieve

Deploy



# Component Example: Application Components

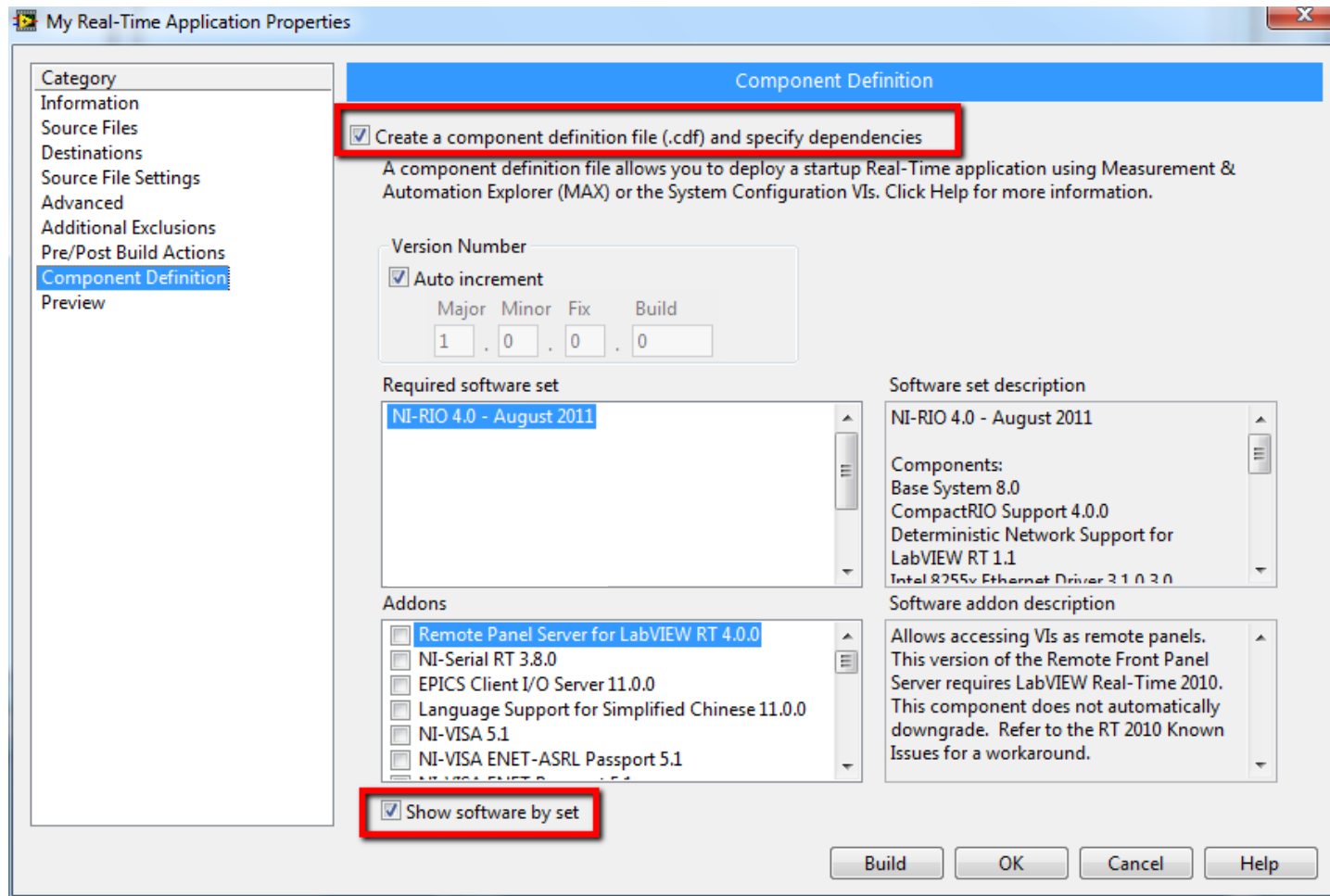
- RT Application is built into a system component
- Allows specification of driver dependencies for application
- Limitation: Project configurations cannot be deployed



- See [White Paper: Using Application Components to Deploy LabVIEW Real-Time Applications](#)

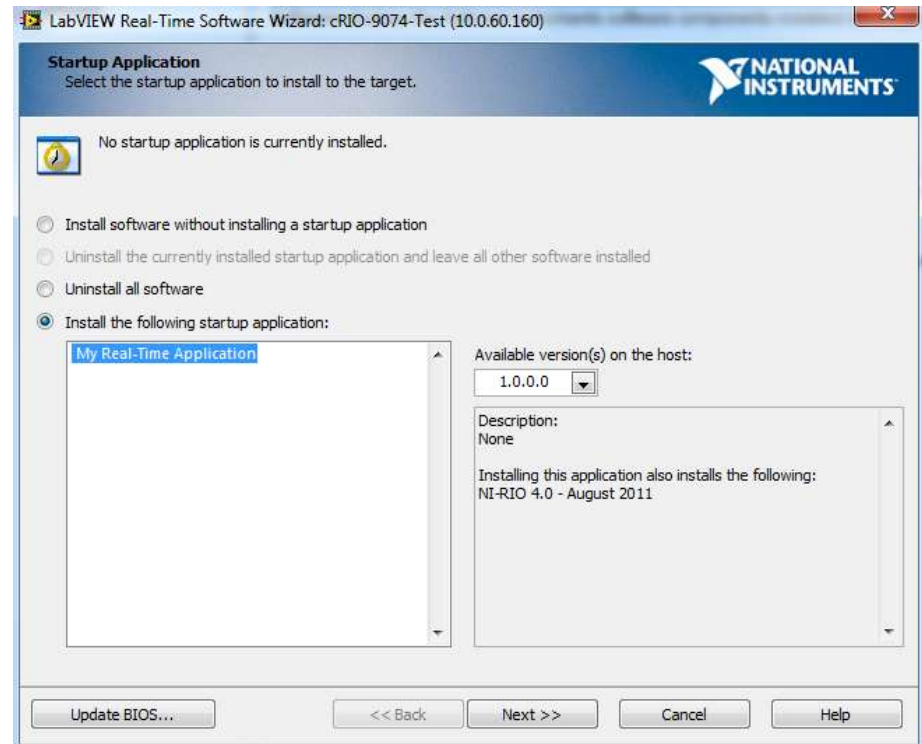
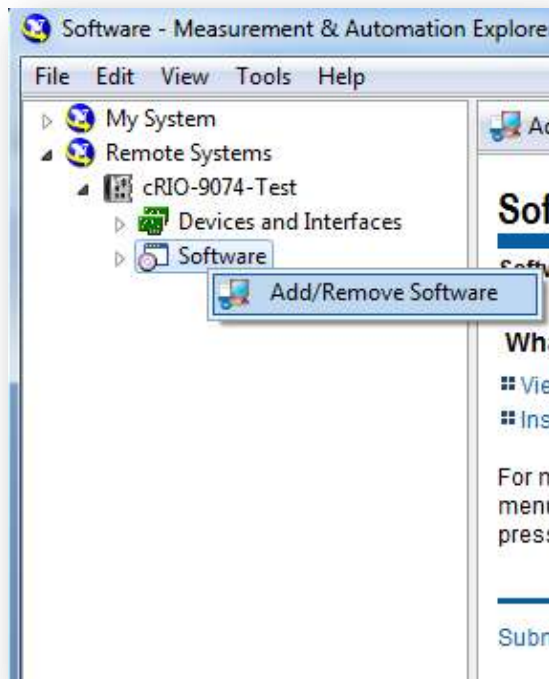


# Application Components



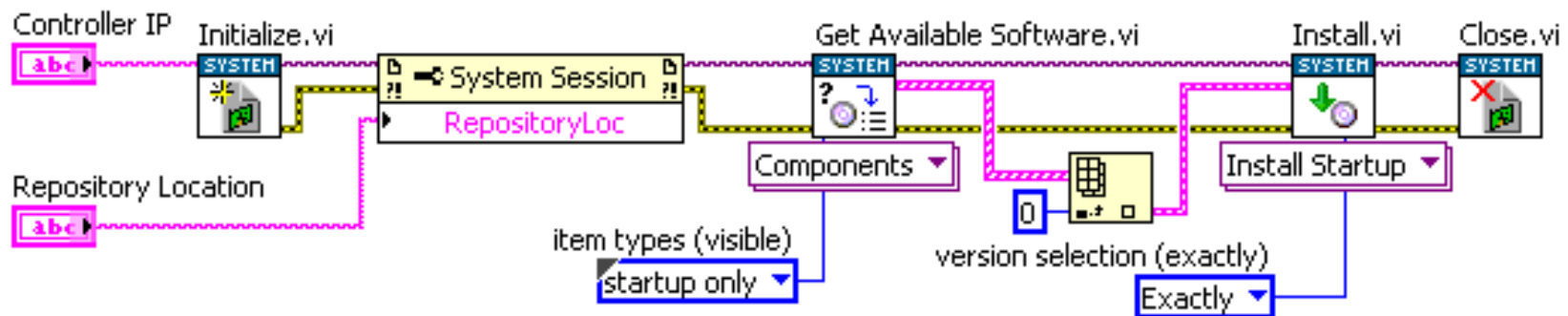
# Application Components Deployment

- MAX works as a GUI to install components to targets



# Application Components Deployment

- System Configuration API to programmatically deploy
- Component Definition File
  - .../National Instruments/RT Images/User Components/[*Project Name*]/[*Target Name*]/[*Build Specification Name*]/[*Version #*]
  - Distribute *RT Images* Folder to other machines



Demo

# APPLICATION COMPONENTS

# Push Deployment Summary

## Advantages

Control

Simplicity

Availability

## Disadvantages

Less reliability

Less scalable

Less secure

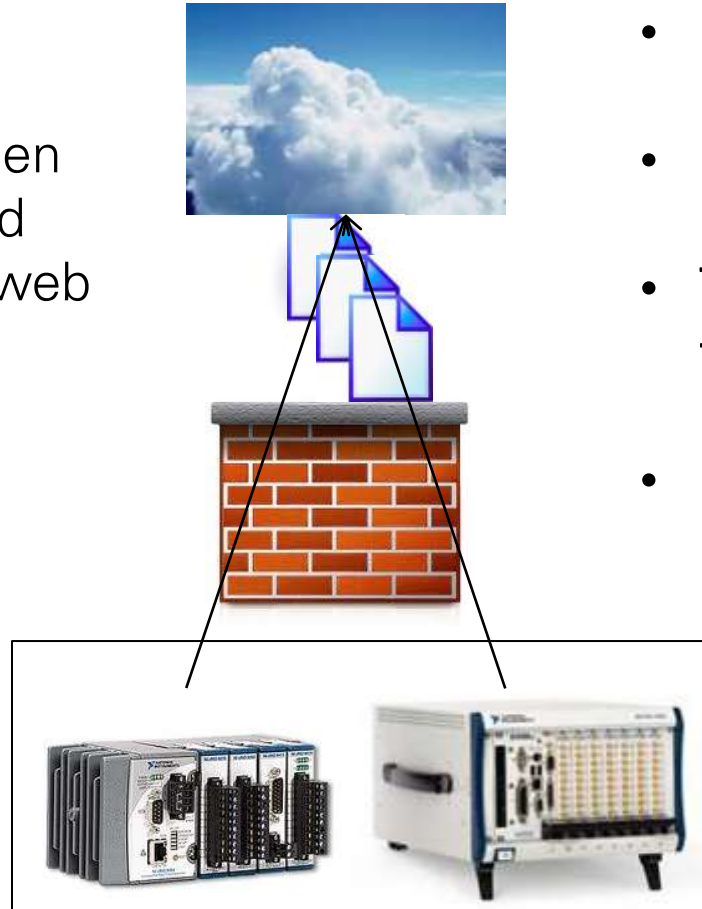
# Pull Model

## Pros:

- Scalable
  - No discovery
- Target controls when updates are applied
- Server could be a web service
- Host not required

## Ex:

- NI Update Service
- Windows Update



## Cons:

- More complex than push
- Requires more infrastructure
- **Target must be able to install updates to itself**
- Custom solution required

# Operating System Support

## Phar Lap/ ETS

- Limited pull update support
- Do not have full write access when running (only safe mode)
- Cannot use NI APIs to install software to itself

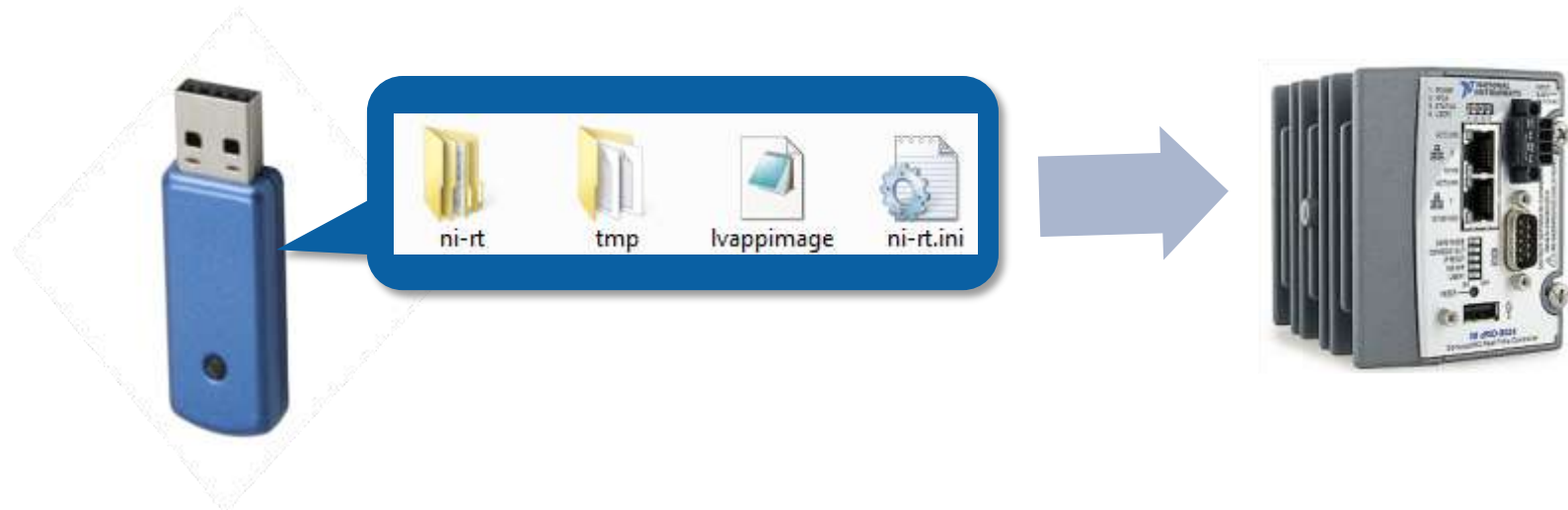
## VxWorks

- Moderate pull update support
- Has write access when running
- Cannot use NI APIs to install software to itself

## NI Linux RT

- Best pull update support
- NI supported API for imaging

# Simple Example: Deploying Image Updates using a USB Memory Device

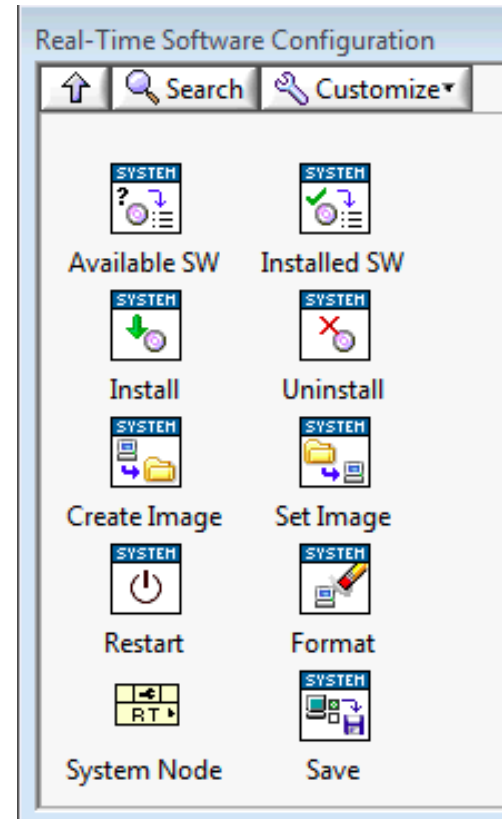


- See [Example: Reference Design for Plugin and Image Deployment to a CompactRIO using a Memory Device](#)



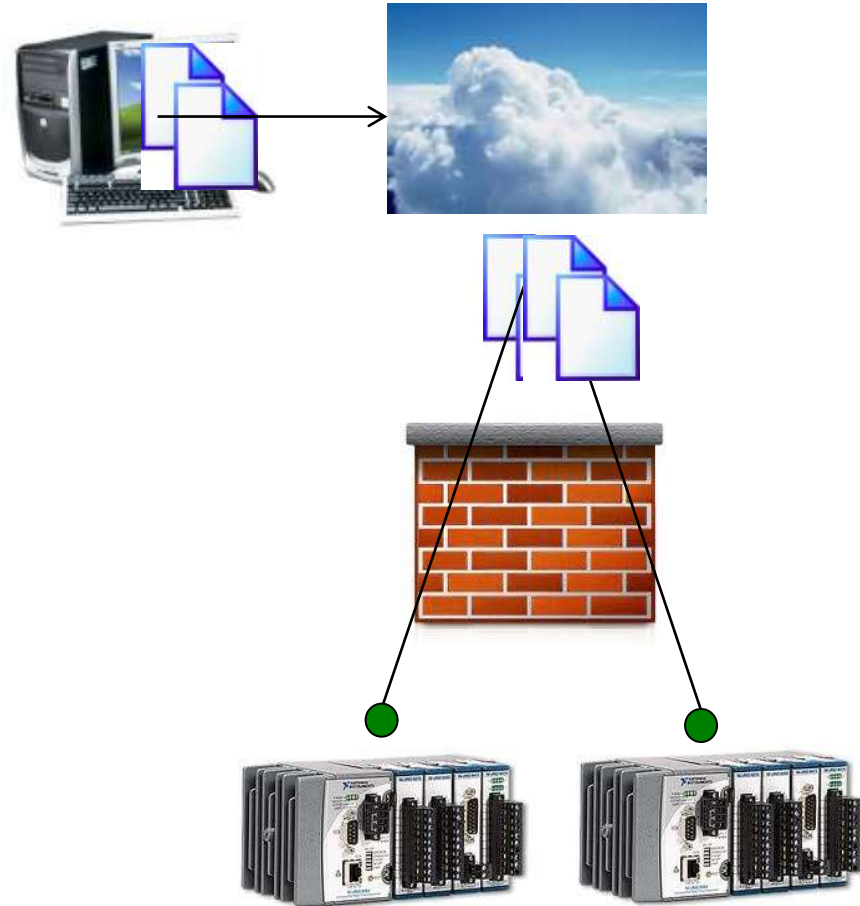
# New Pull Based System Updates on NI Linux Real Time

- NI Linux Real-Time targets can directly call "Set Image"
  - Enables targets to reimage themselves
  - Images can be pulled down from the network or stored on a USB drive
- Specify additional metadata when creating an RT image (title, version, description)
- Blacklist Wildcards with System Imaging
  - Globbing: "\*" and "?"
  - Character set matches: [abc]



# Pull based Architecture

- Clients and targets talk to the server
  - Clients don't directly talk to the targets
- Benefits:
  - Clients upload updates directly to the server
  - Clients configure which targets need to receive the update on the server
  - Targets pull updates directly from the server
  - Targets provide progress information to the server
  - Communication to the server can be secured



# Considerations for Pull Deployment



What do we store on our server? Who has access?

How are these updates obtained? Where are they stored?



What updates can a target pull down?

When and how do we determine if there are updates?



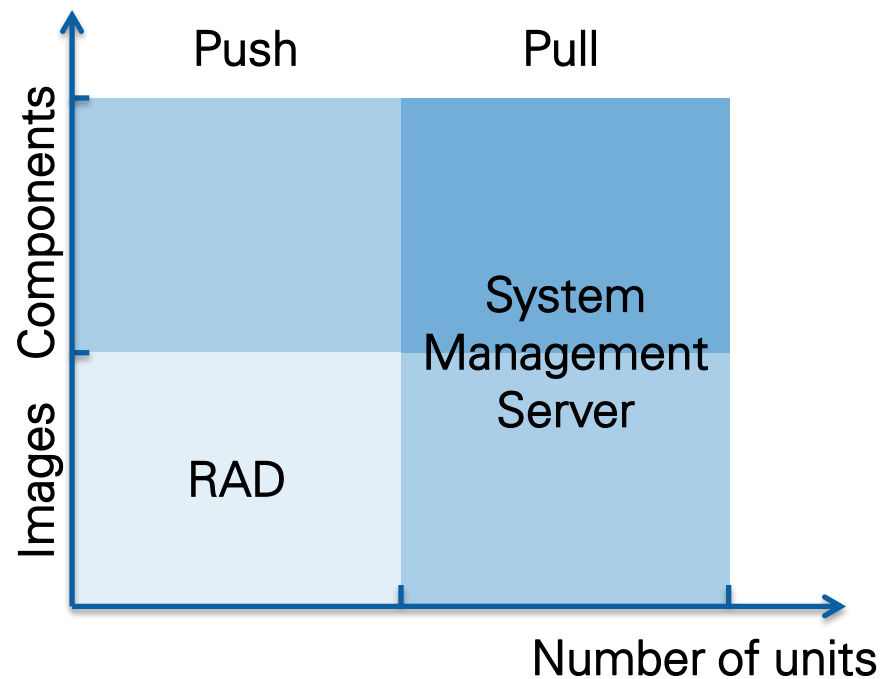
How and when are the updates applied?

# Push vs. Pull

Push	Pull
<ul style="list-style-type: none"><li>• Simpler implementation</li></ul>	<ul style="list-style-type: none"><li>• Control over deployment time and safe state</li></ul>
<ul style="list-style-type: none"><li>• Code updates managed by single host</li></ul>	<ul style="list-style-type: none"><li>• Better for N:N deployments</li></ul>
<ul style="list-style-type: none"><li>• More support from existing NI Technologies</li></ul>	<ul style="list-style-type: none"><li>• Better for automated target management</li></ul>

# Conclusion

- Deployment is a complex topic, and there is not a “one-size-fits all” solution
- NI recognizes system management and deployment as a key challenge facing developers today
- NI is continuing to invest in solutions to make deployment easier



# Questions?