

# The ATLAS RunTimeTester Software

Krzysztof Ciba,  
Alexander Richards,  
Peter Sherwood,  
**Brinick Simmons**

CHEP 2009, Prague  
Software Components session



# Introduction

“A Python-coded  
framework for testing  
ATLAS software”

# Introduction

➤ Why does the RTT exist?

## The Problem

- ATLAS has a large quantity of complex, changing code
- Results can alter due to changes in libraries used by the code, not only due to changes in the code itself

# Introduction

➤ Why does the RTT exist?

## The Problem

- ATLAS has a large quantity of complex, changing code
- Results can alter due to changes in libraries used by the code, not only due to changes in the code itself

## The RTT provides a tool that:

- Automates-away the pain of manual testing

## The Problem

- ATLAS has a large quantity of complex, changing code
- Results can alter due to changes in libraries used by the code, not only due to changes in the code itself

## The RTT provides a tool that:

- Automates-away the pain of manual testing
- Tests daily if developer code runs, and if results remain consistent in time by providing integration with:
  - DCube regression test framework<sup>[1]</sup>
  - CoolDozer long-term trend monitoring framework<sup>[2]</sup>

[1] "DCube", Krzysztof Ciba, <https://twiki.cern.ch/twiki/bin/view/Sandbox/DCubeDoc>

[2] "CoolDozer", Krzysztof Ciba, <https://twiki.cern.ch/twiki/bin/view/Main/CoolDozer>

# Introduction

➤ Why does the RTT exist?

## The Problem

- ATLAS has a large quantity of complex, changing code
- Results can alter due to changes in libraries used by the code, not only due to changes in the code itself

## The RTT provides a tool that:

- Automates-away the pain of manual testing
- Tests daily if developer code runs, and if results remain consistent in time by providing integration with:
  - DCube regression test framework<sup>[1]</sup>
  - CoolDozer long-term trend monitoring framework<sup>[2]</sup>
- Provides access to other (ATLAS and External) tools that facilitate comprehension and presentation of test results

[1] "DCube", Krzysztof Ciba, <https://twiki.cern.ch/twiki/bin/view/Sandbox/DCubeDoc>

[2] "CoolDozer", Krzysztof Ciba, <https://twiki.cern.ch/twiki/bin/view/Main/CoolDozer>

## The Problem

- ATLAS has a large quantity of complex, changing code
- Results can alter due to changes in libraries used by the code, not only due to changes in the code itself

## The RTT provides a tool that:

- Automates-away the pain of manual testing
- Tests daily if developer code runs, and if results remain consistent in time by providing integration with:
  - DCube regression test framework<sup>[1]</sup>
  - CoolDozer long-term trend monitoring framework<sup>[2]</sup>
- Provides access to other (ATLAS and External) tools that facilitate comprehension and presentation of test results
- Provides a centralised organisation for release coordinators to ascertain the state of a particular ATLAS release

[1] "DCube", Krzysztof Ciba, <https://twiki.cern.ch/twiki/bin/view/Sandbox/DCubeDoc>

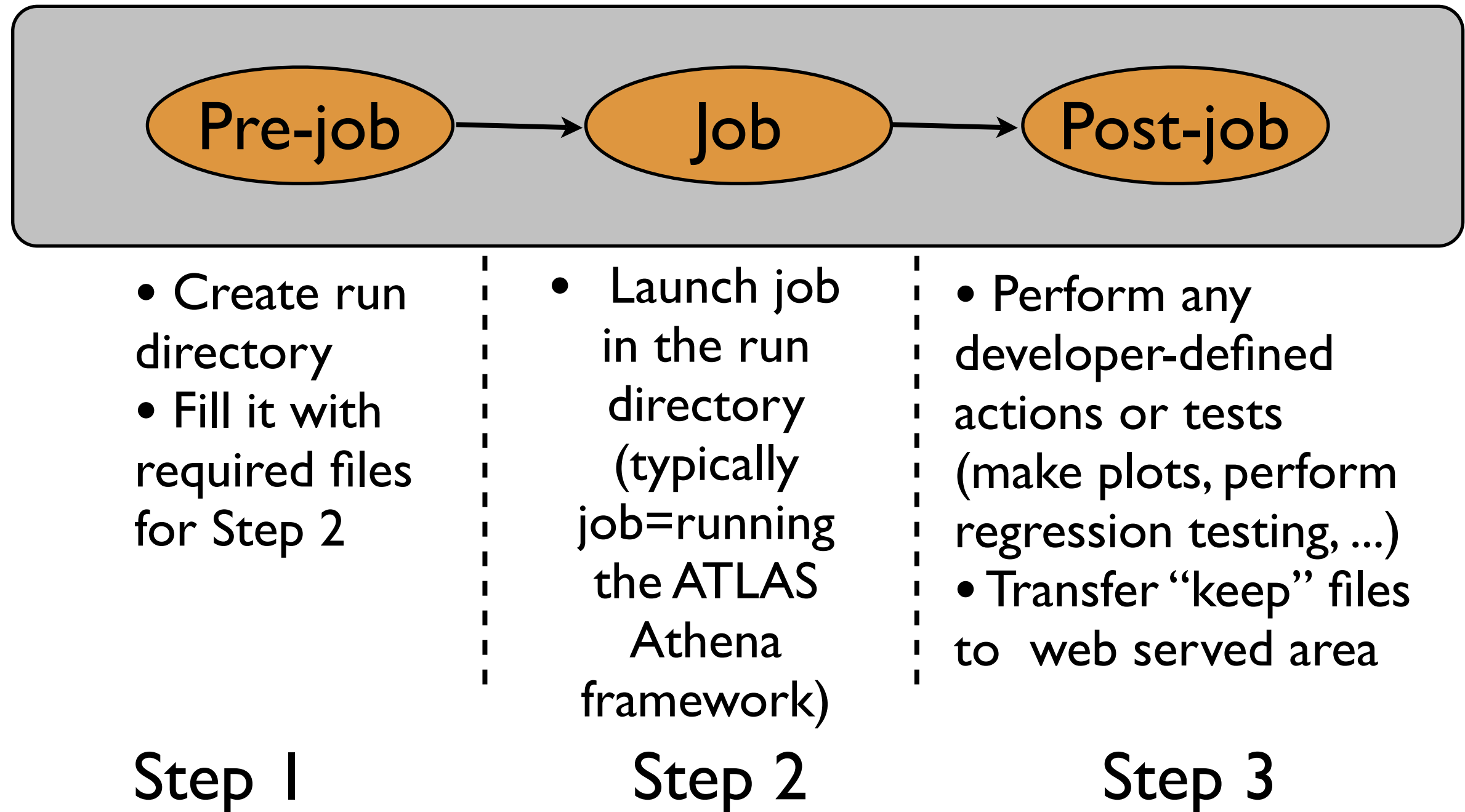
[2] "CoolDozer", Krzysztof Ciba, <https://twiki.cern.ch/twiki/bin/view/Main/CoolDozer>



# Introduction

## ➤ RTT jobs

- The RTT runs RTT jobs
- An RTT job is a three step process:



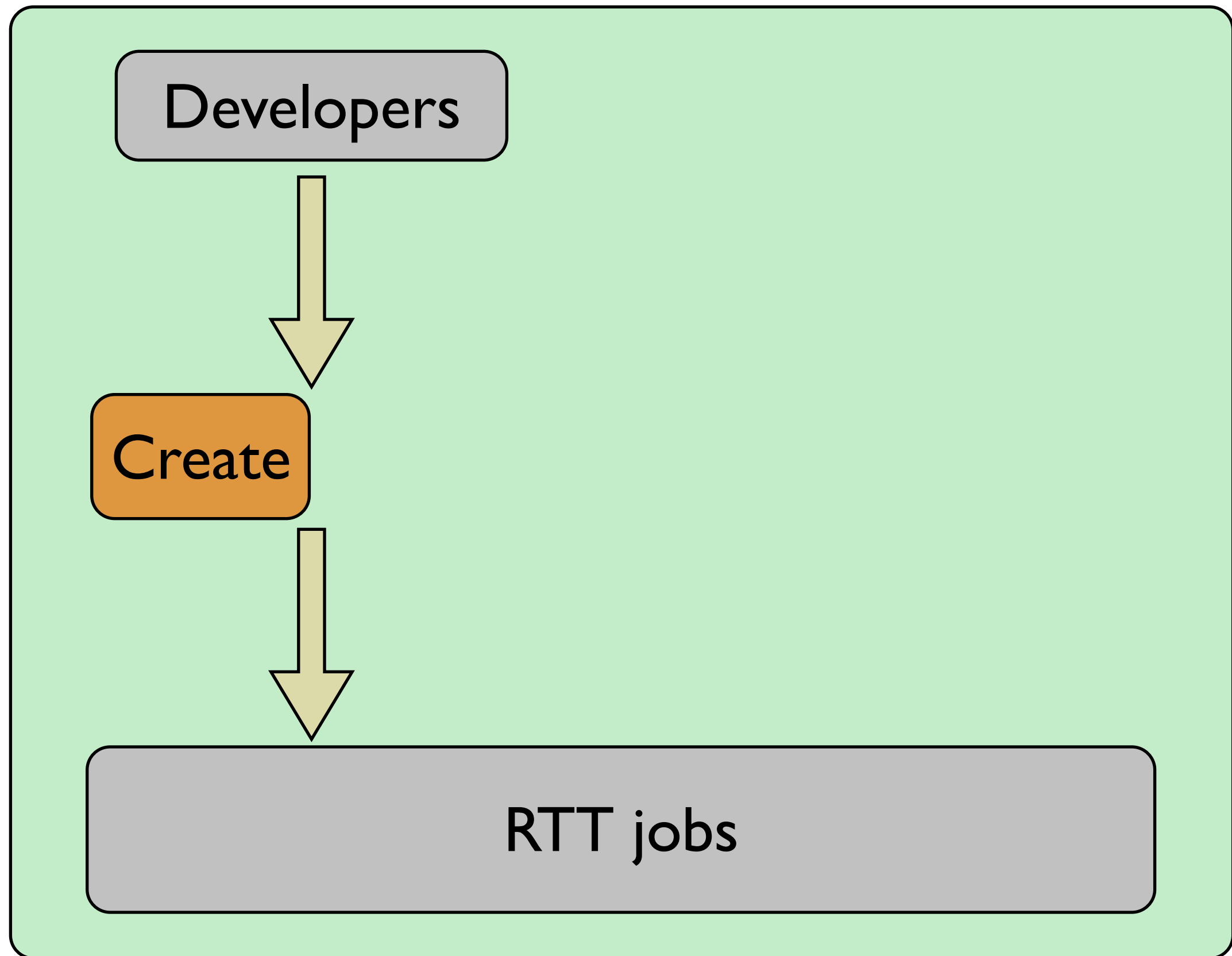
# Introduction

- The RTT and ATLAS developer interaction model



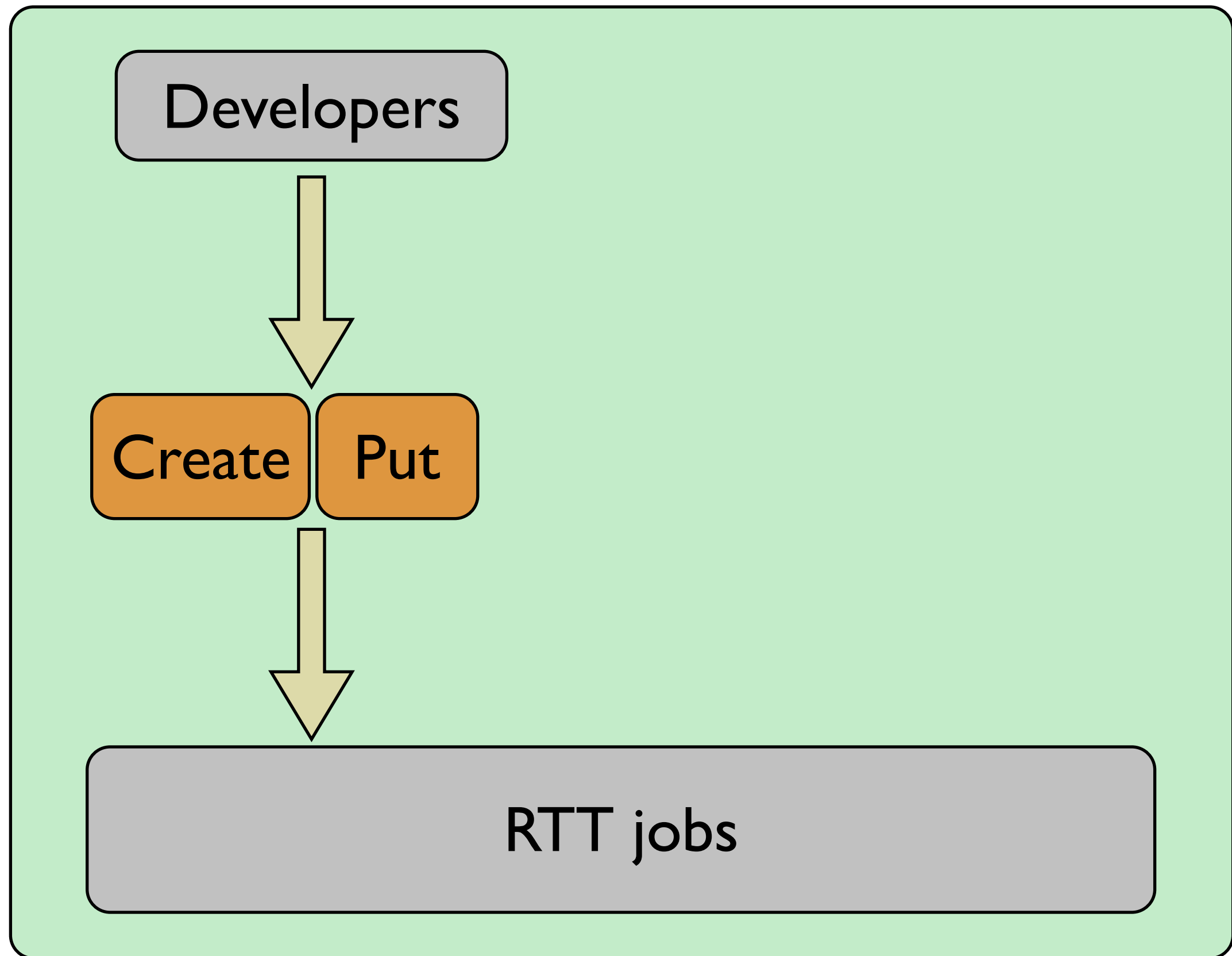
# Introduction

➤ The RTT and ATLAS developer interaction model



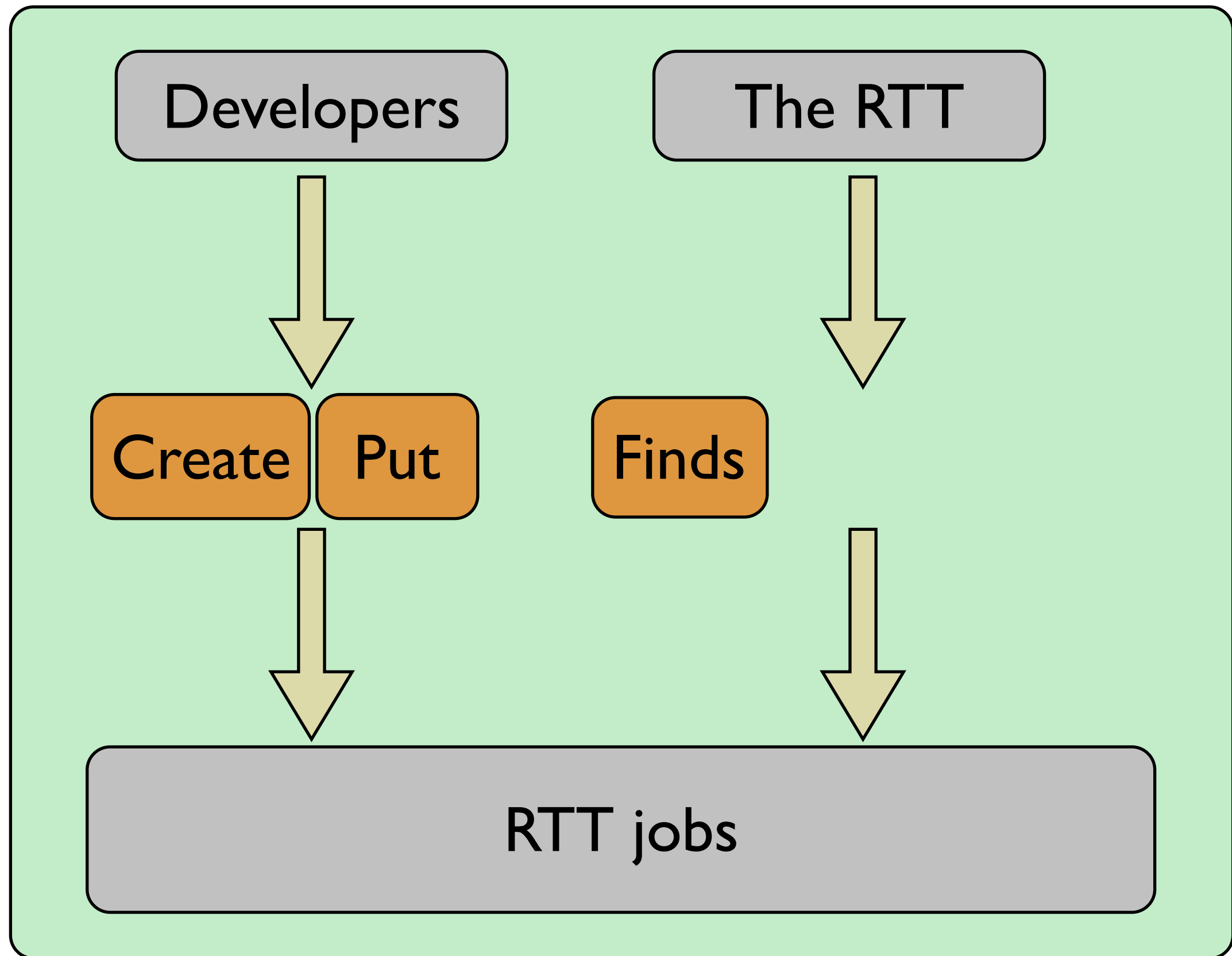
# Introduction

➤ The RTT and ATLAS developer interaction model



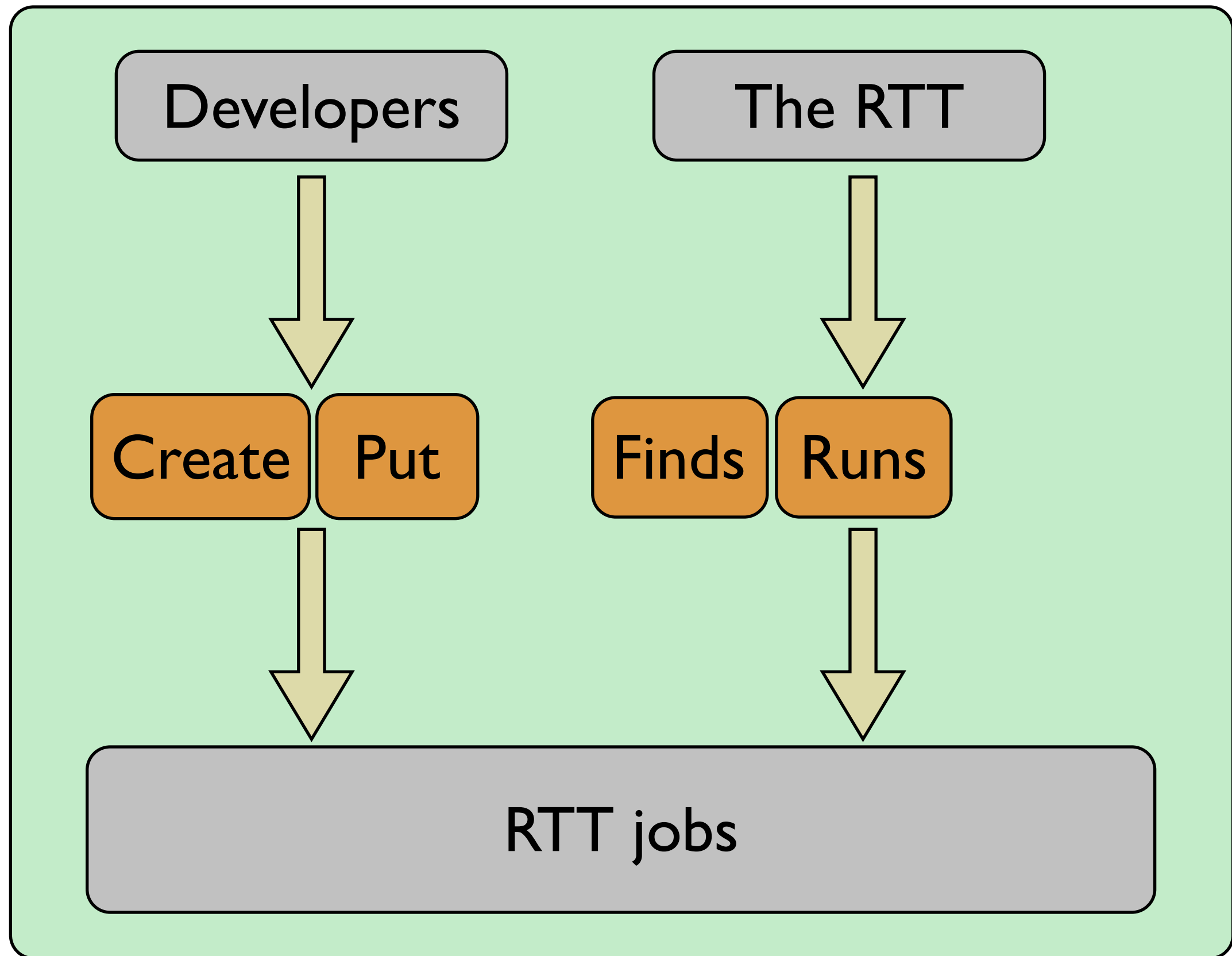
# Introduction

➤ The RTT and ATLAS developer interaction model



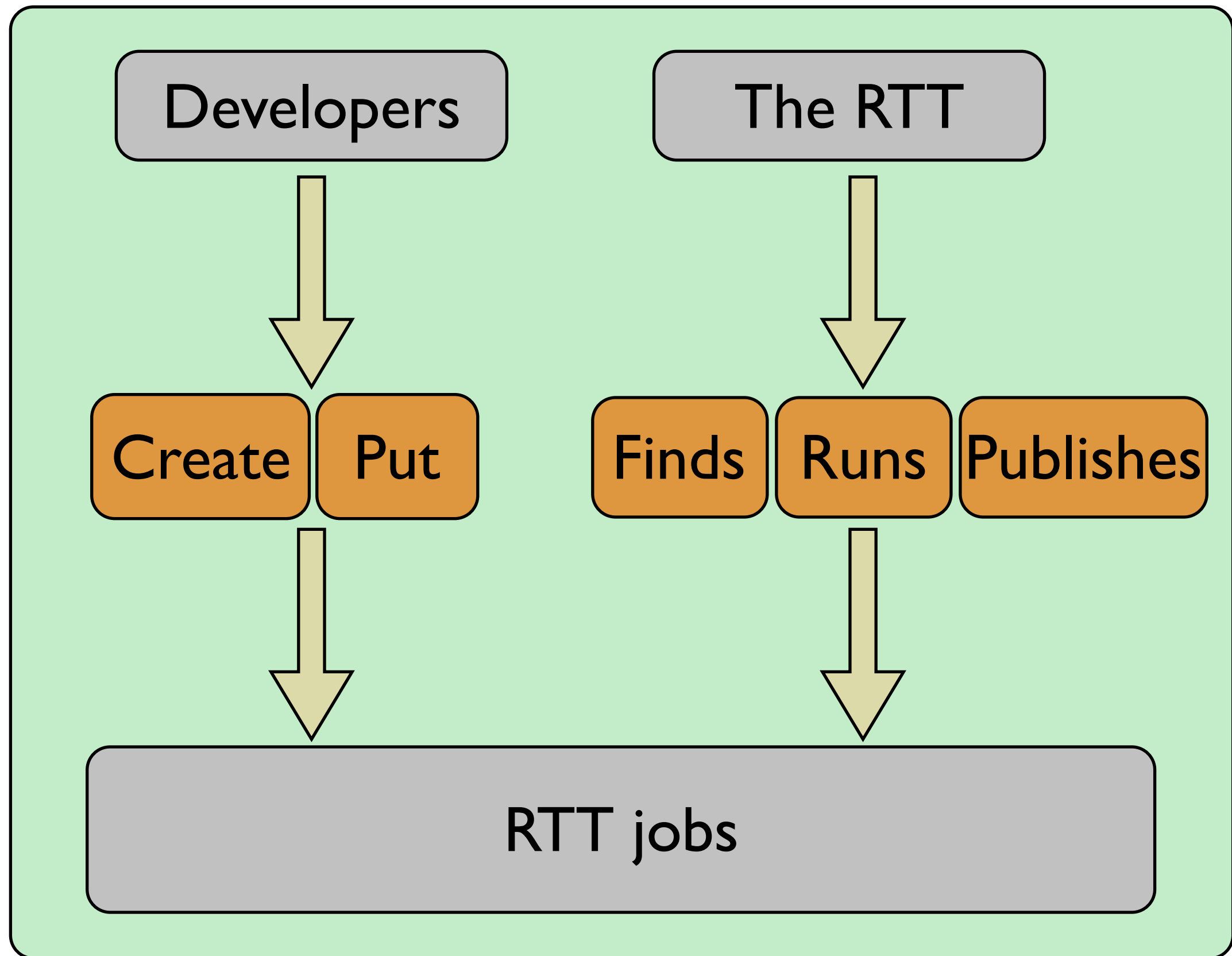
# Introduction

➤ The RTT and ATLAS developer interaction model



# Introduction

➤ The RTT and ATLAS developer interaction model



# Developers

Create



Within their package, a developer must:

- Create a unified test configuration XML file: this is where they define the RTT jobs to run
- Add 2 “magic” lines to their existing CMT<sup>[1]</sup> requirements file that indicate:
  - ▶ This package has RTT jobs to run
  - ▶ The location of the XML file within the package

[1] CMT: Code Management Tool, <http://www.cmts.site.org>

# Developers

➤ Create :: the Unified Test Configuration XML file

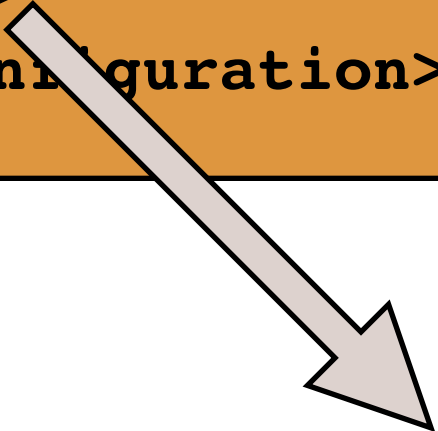
```
<?xml version="1.0"?>  
<unifiedTestConfiguration>  
  <atn>...</atn>  
  <kv>...</kv>  
  <rtt>...</rtt>  
</unifiedTestConfiguration>
```

“Unified” because  
same file used to talk  
to any of the 3 ATLAS  
test frameworks

# Developers

➤ Create :: the Unified Test Configuration XML file

```
<?xml version="1.0"?>
<unifiedTestConfiguration>
  <atn>...</atn>
  <kv>...</kv>
  <rtt>...</rtt>
</unifiedTestConfiguration>
```



“Unified” because  
same file used to talk  
to any of the 3 ATLAS  
test frameworks

RTT section: 3 parts

```
<rtt>
  <mailto>...</mailto>
  <jobList>...</jobList>
  <jobGroups>...</jobGroups>
</rtt>
```

# Developers

➤ Create :: the Unified Test Configuration XML file

```
<mailto></mailto>
```

- RTT sends mail here if there is job/test failure

# Developers

## ➤ Create :: the Unified Test Configuration XML file

```
<mailto></mailto>
```

- RTT sends mail here if there is job/test failure

```
<jobList>  
  [job1]  
  [job2]  
  ....  
  ....  
</jobList>
```

- No limit on the number of jobs
- Here “job” is step 2 of an RTT job. Typically it involves running some part of the ATLAS Athena software

# Developers

## ➤ Create :: the Unified Test Configuration XML file

```
<mailto></mailto>
```

- RTT sends mail here if there is job/test failure

```
<jobList>  
  [job1]  
  [job2]  
  ....  
  ....  
</jobList>
```

- No limit on the number of jobs
- Here “job” is step 2 of an RTT job. Typically it involves running some part of the ATLAS Athena software

```
<jobGroups>  
  [jobGroup1]  
  [jobGroup2]  
  ....  
  ....  
</jobGroups>
```

- No limit on the number of job groups
- Job groups define pre/post-job actions/tests to execute
- An RTT job must belong to one (and only one) job group
- All RTT jobs in the same group get the same treatment
- Groups must define a parent group
- Thus: RTT constructs a job group “tree” with a “Top” group, defined by RTT, which provides common behaviour inherited by all child job groups

# Developers

Put

- A developer has added RTT jobs to his package
- Now what, where does he put his package?
- There are exactly two choices:
  - ▶ Outside an ATLAS nightly release
  - ▶ Inside an ATLAS nightly release



## Outside the release

### *Manual, interactive running*

- Run manually by the developer on their Linux box
- Typically one-few jobs
- Provides small-scale tests of a developer's locally-based package
- Used to validate, say, some new functionality in the package before committing/tagging

# Developers

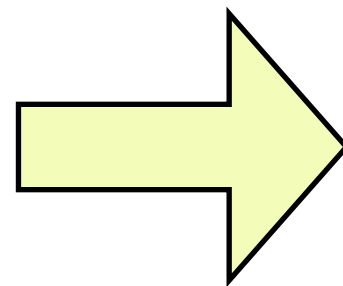
➤ Put :: RTT operative modes

Inside the release

*Automatic, batch running*

- Run daily @ CERN via acron on dedicated launch nodes
- Typically 100's of jobs
- Provides larger-scale testing of an ATLAS nightly build
- Used to validate software components in the build (reconstruction, simulation, ...)

We will look  
only at this  
one



## Inside the release

*Automatic, batch running*

- Run daily @ CERN via acron on dedicated launch nodes
- Typically 100's of jobs
- Provides larger-scale testing of an ATLAS nightly build
- Used to validate software components in the build (reconstruction, simulation, ...)

# The RTT

Finds and Runs

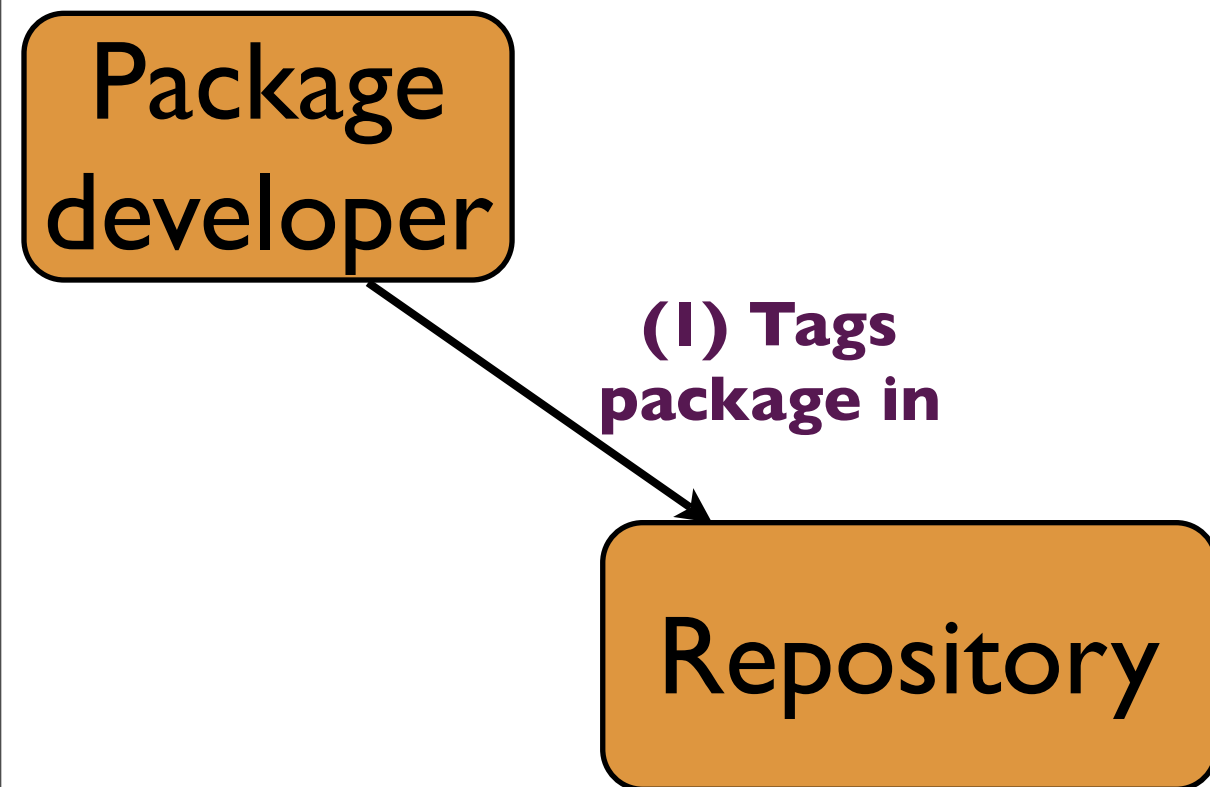
# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

Package  
developer

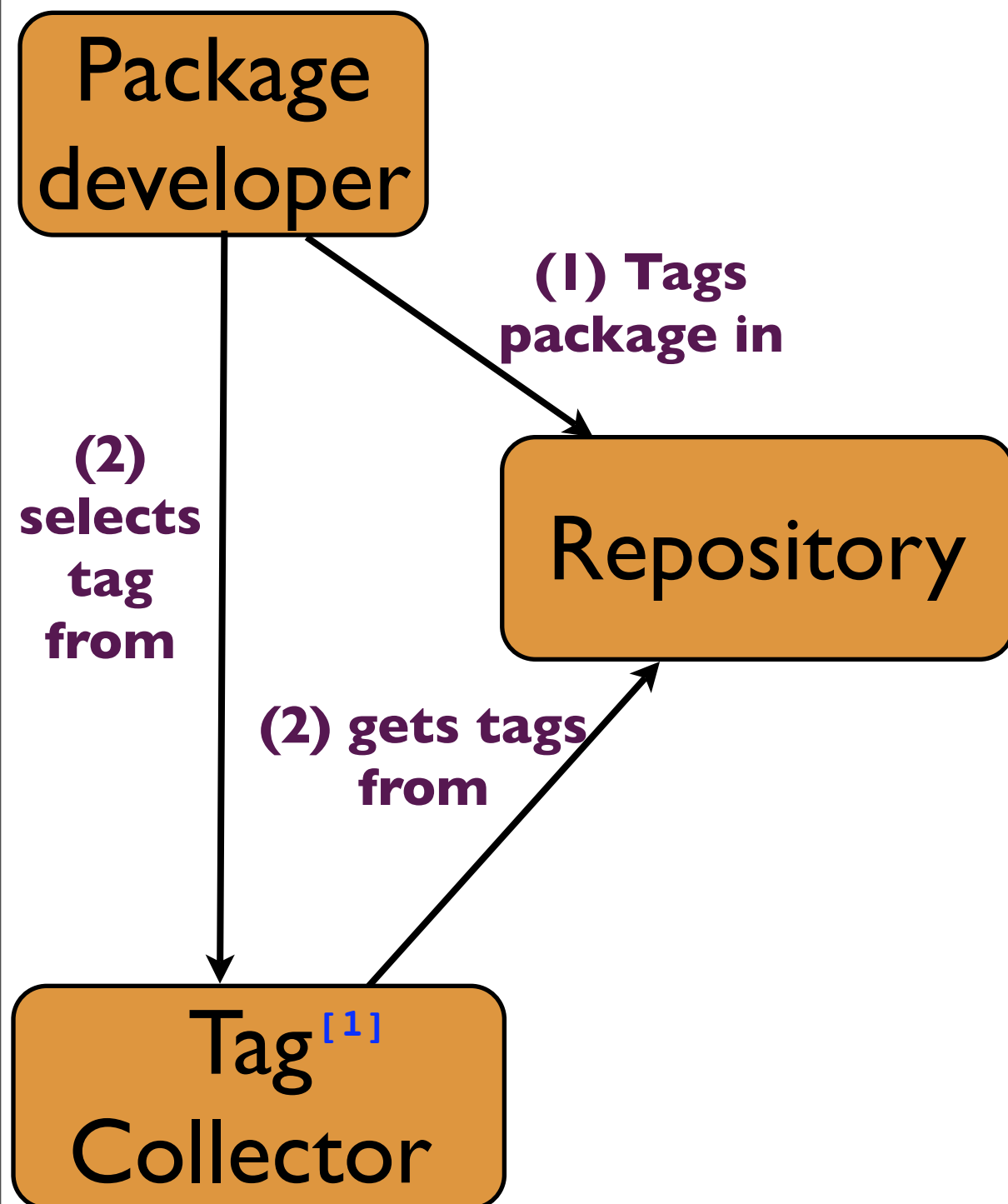
# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation



# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

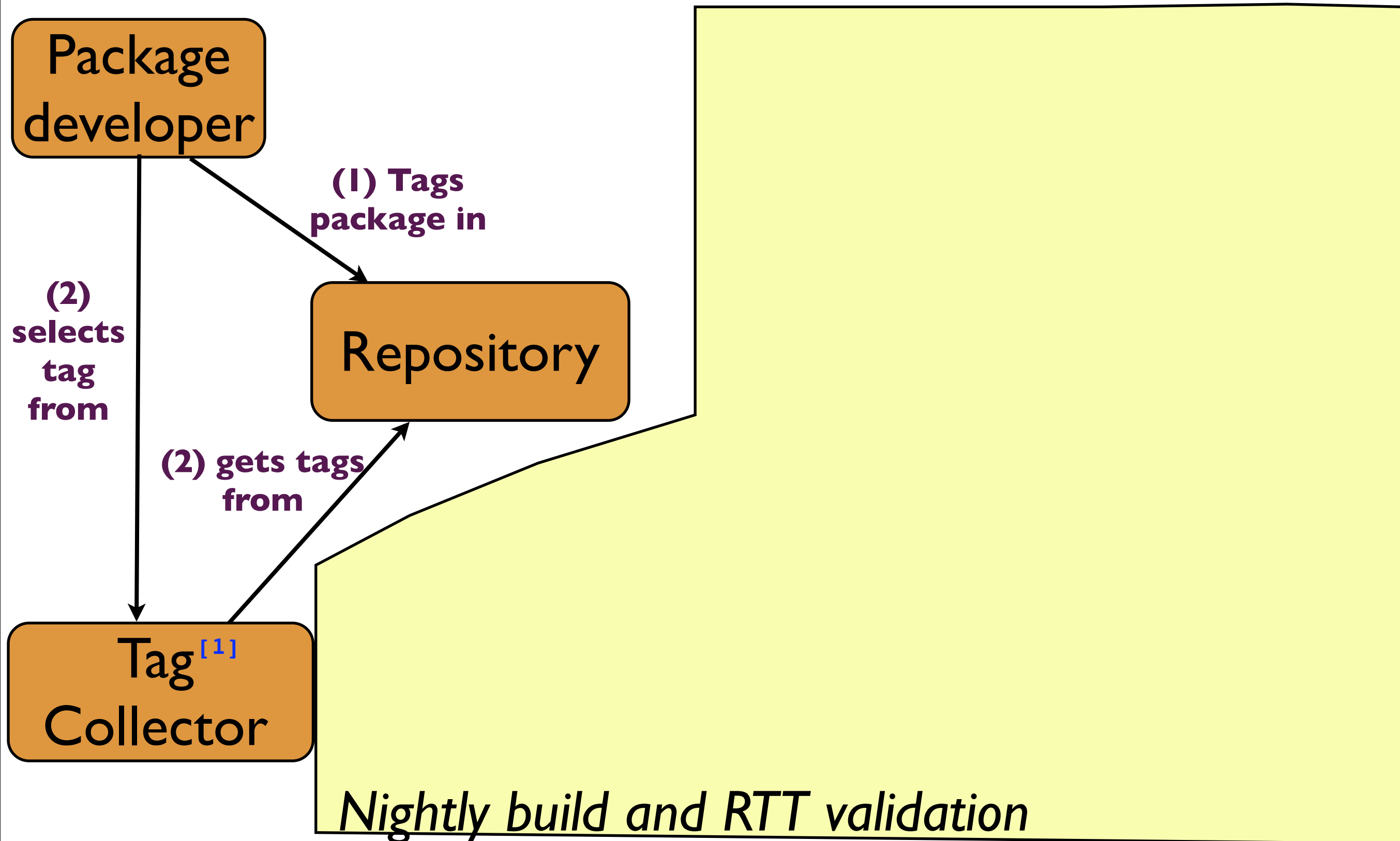


[1] <http://atlastagcollector.in2p3.fr>



# The RTT

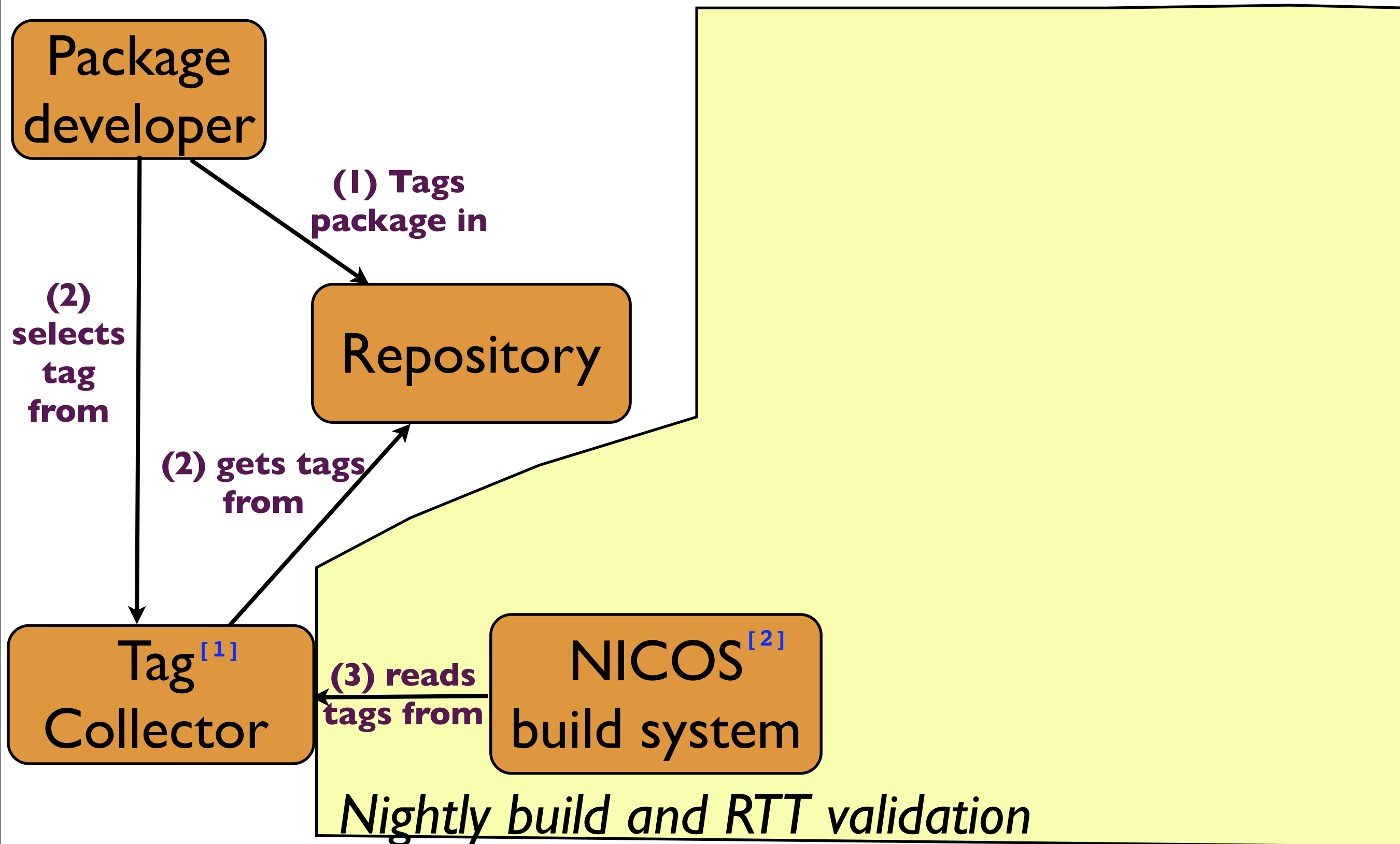
➤ Finds and Runs :: integration within the ATLAS software organisation and validation



[1] <http://atlastagcollector.in2p3.fr>

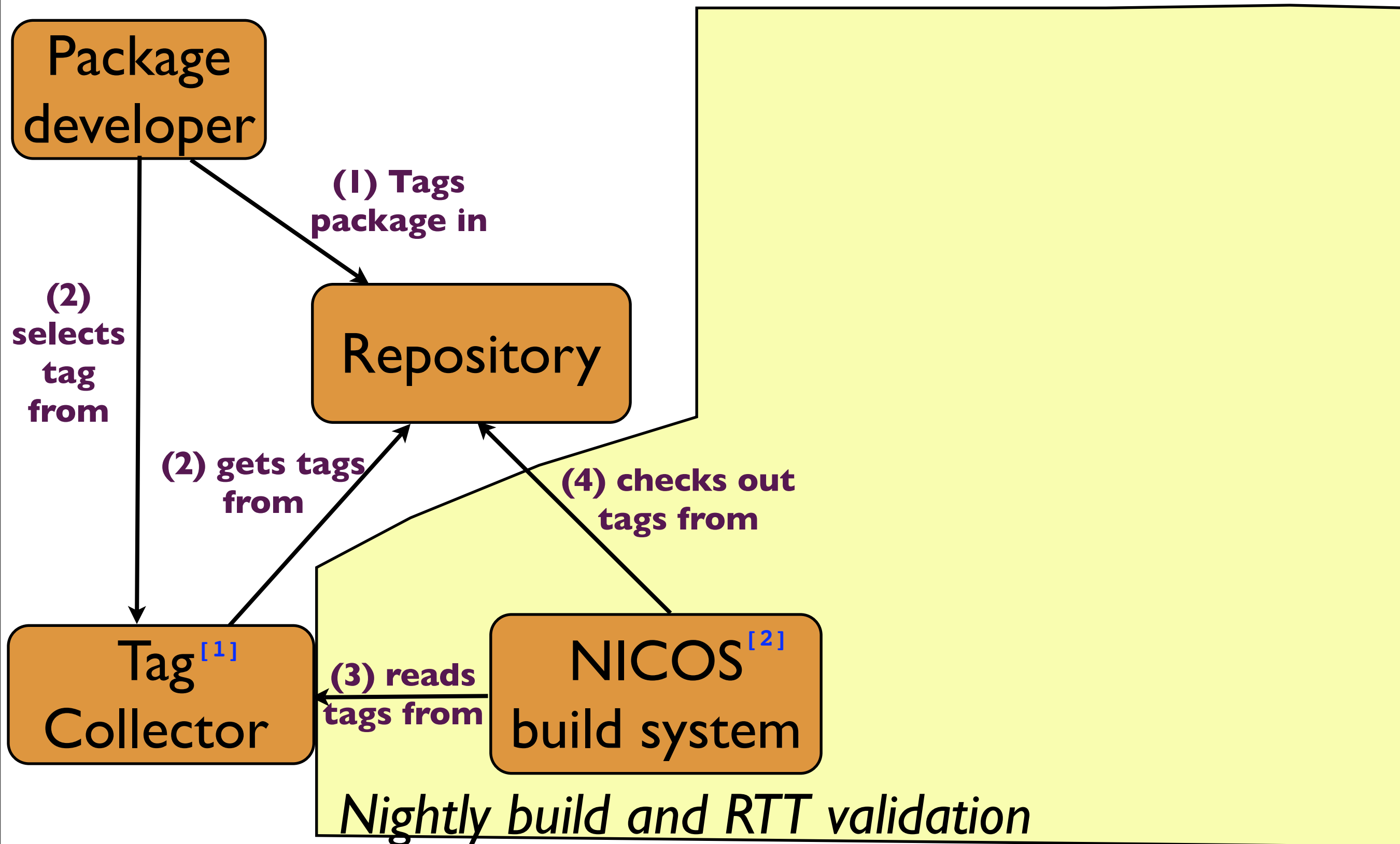
# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation



# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

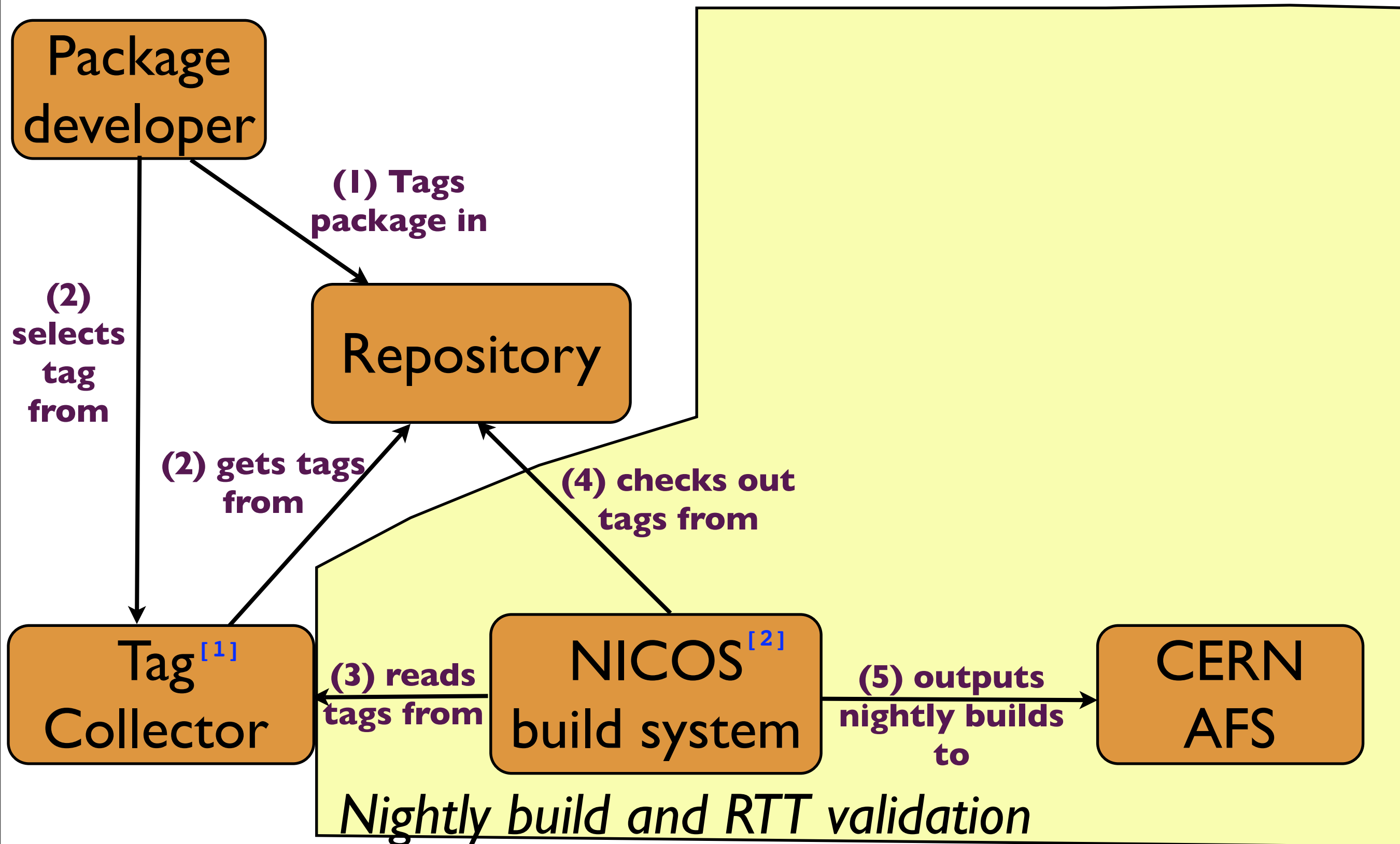


[1] <http://atlastagcollector.in2p3.fr>

[2] A.Undrus et al. "Organisation and Management of ATLAS nightly builds", CHEP 2009

# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

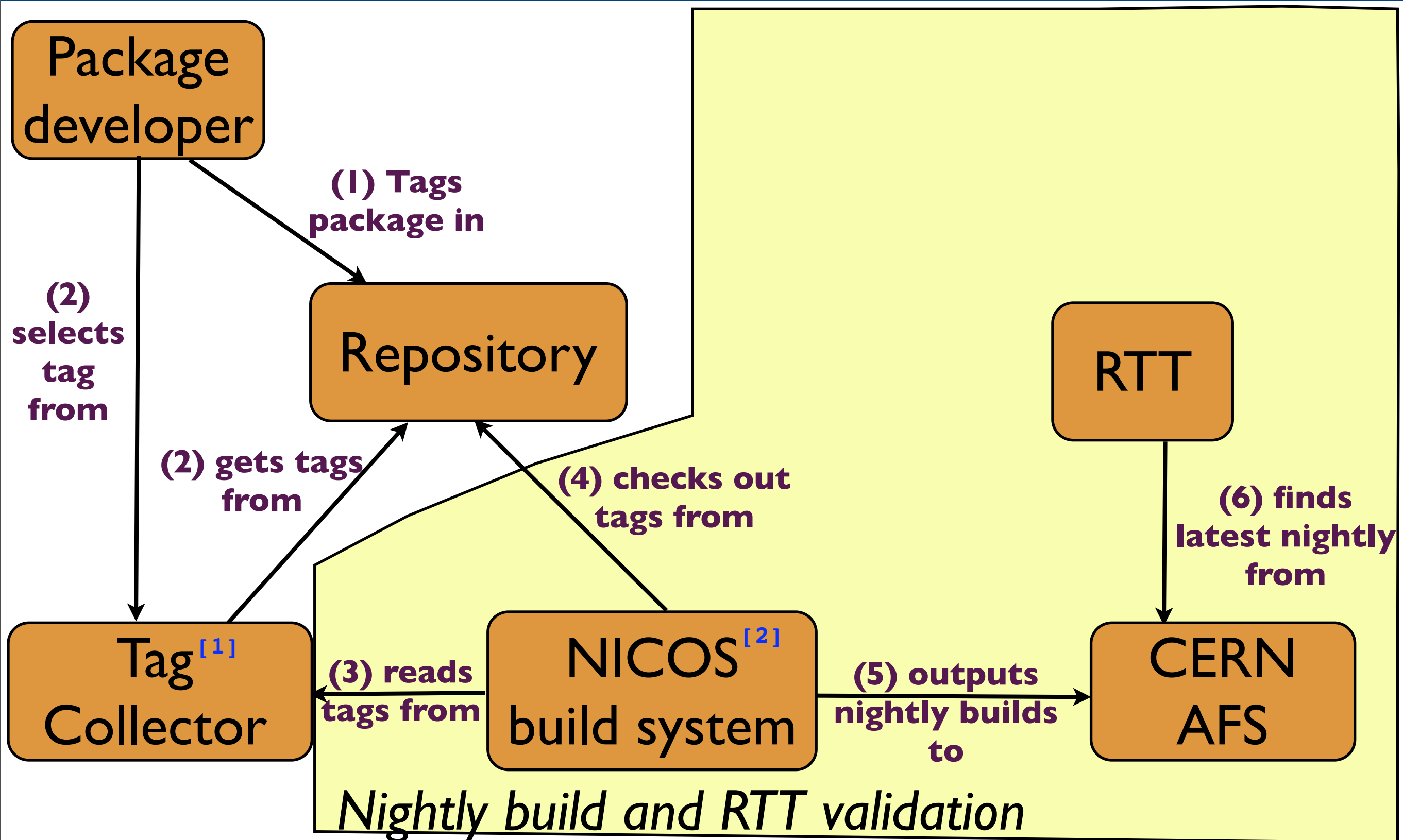


[1] <http://atlastagcollector.in2p3.fr>

[2] A.Undrus et al. "Organisation and Management of ATLAS nightly builds", CHEP 2009

# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

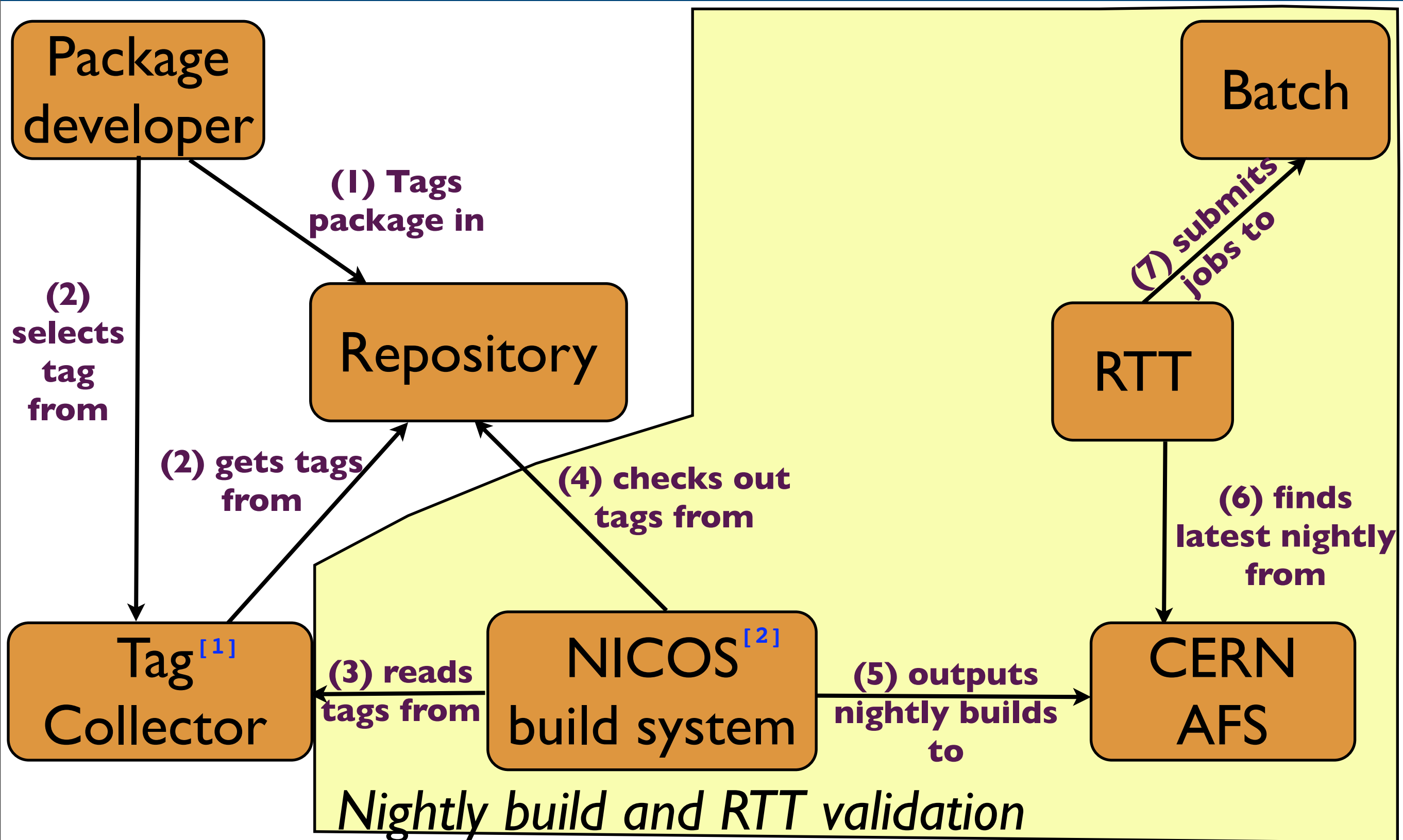


[1] <http://atlastagcollector.in2p3.fr>

[2] A.Undrus et al. "Organisation and Management of ATLAS nightly builds", CHEP 2009

# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

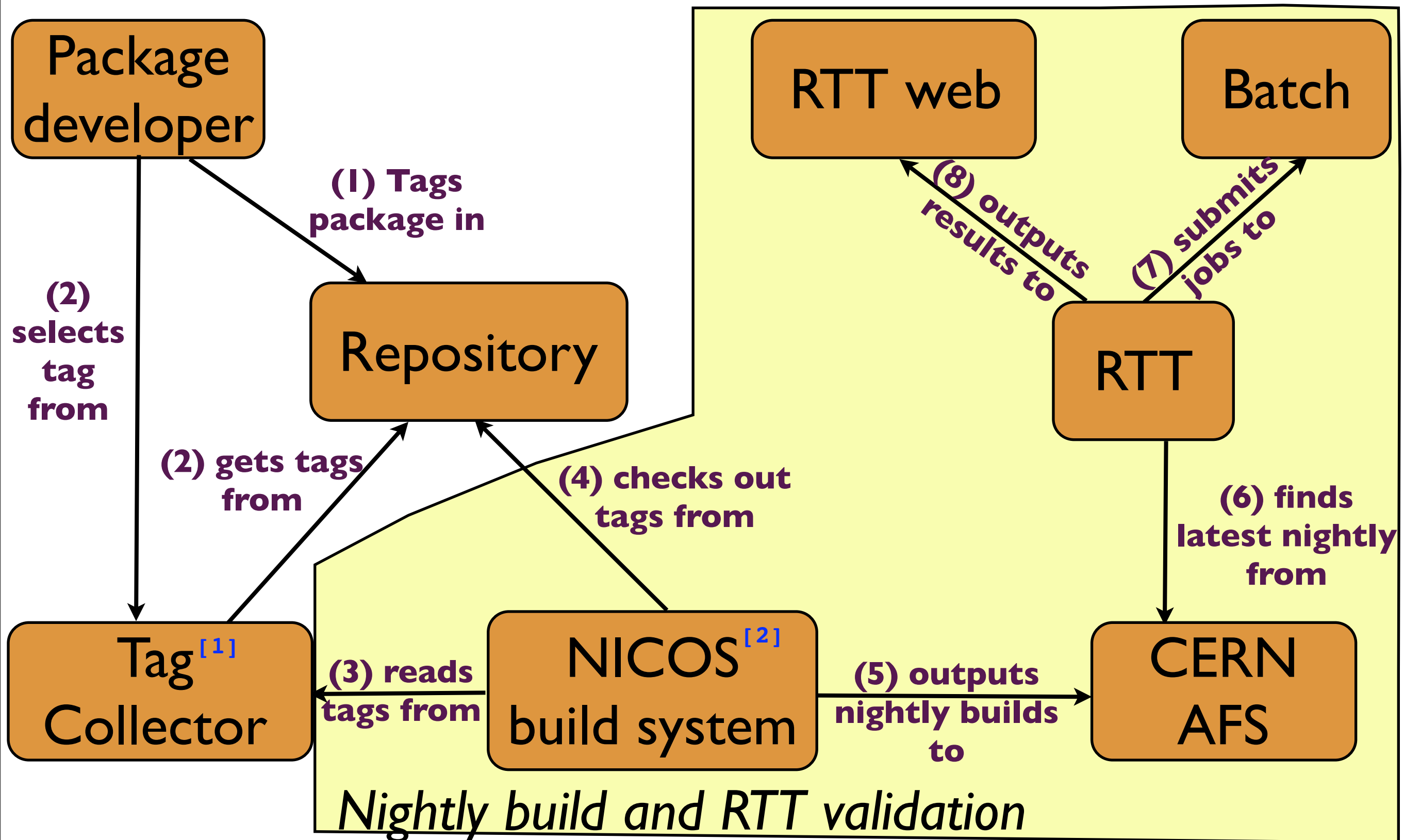


[1] <http://atlastagcollector.in2p3.fr>

[2] A.Undrus et al. "Organisation and Management of ATLAS nightly builds", CHEP 2009

# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

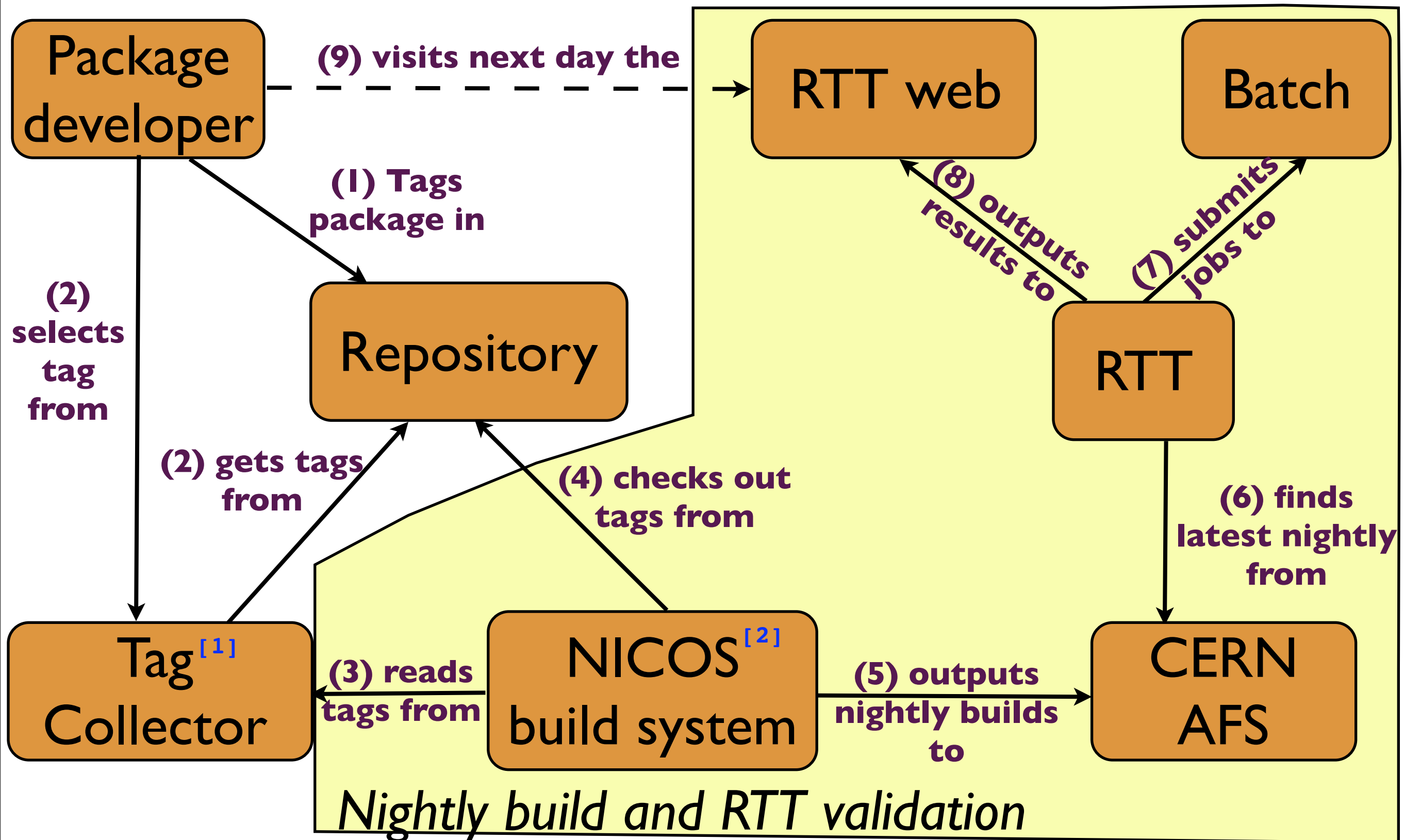


[1] <http://atlastagcollector.in2p3.fr>

[2] A.Undrus et al. "Organisation and Management of ATLAS nightly builds", CHEP 2009

# The RTT

➤ Finds and Runs :: integration within the ATLAS software organisation and validation

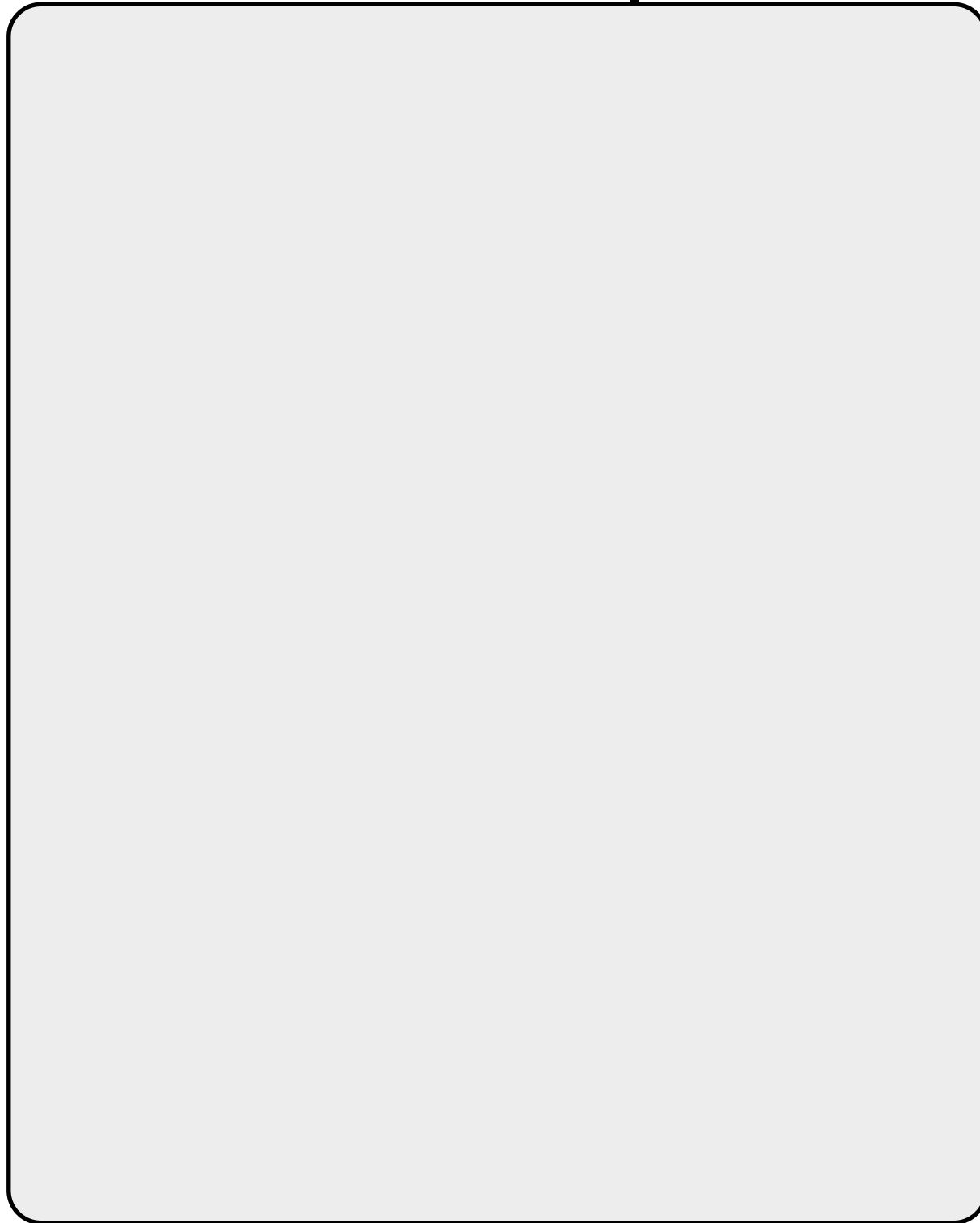


[1] <http://atlastagcollector.in2p3.fr>

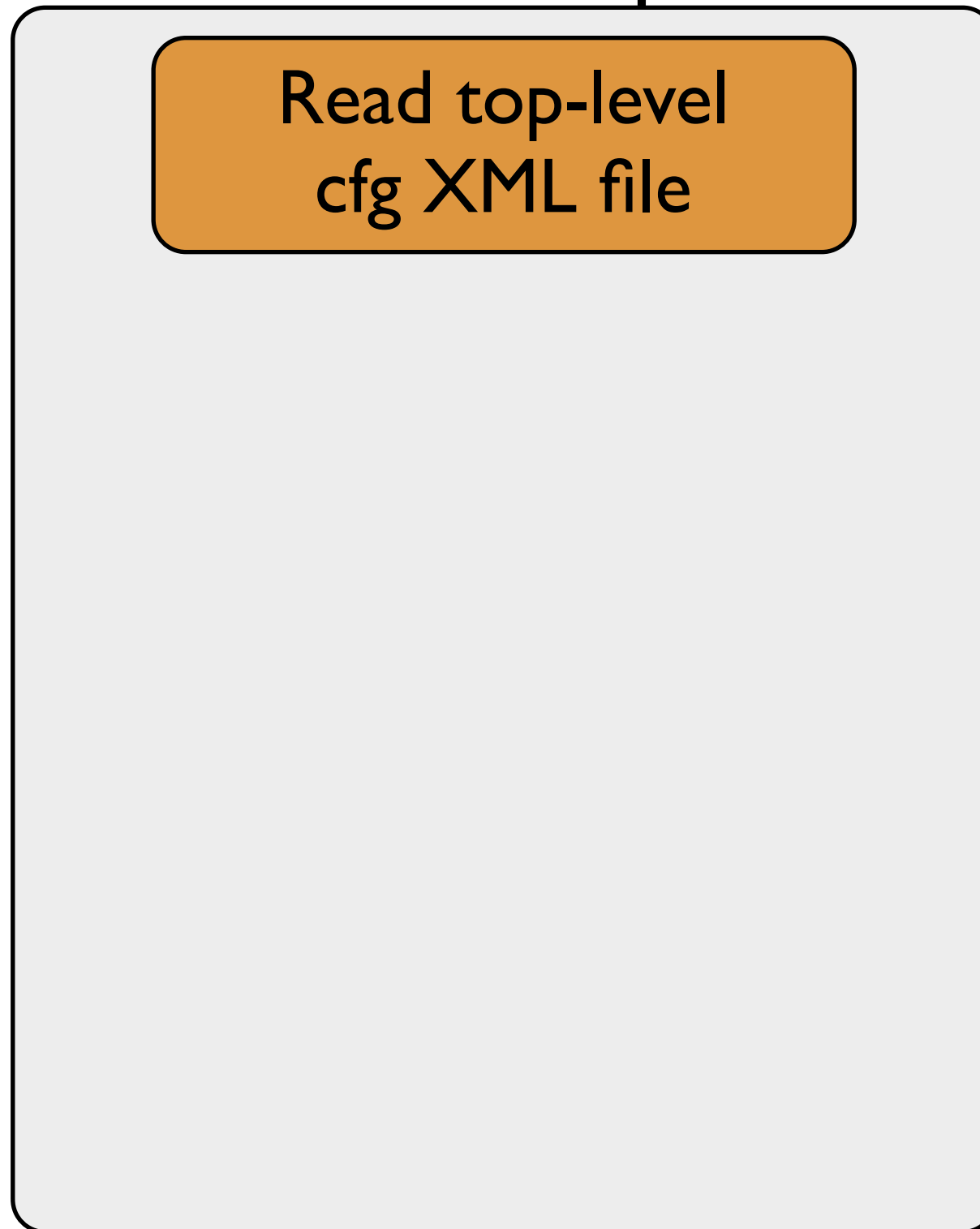
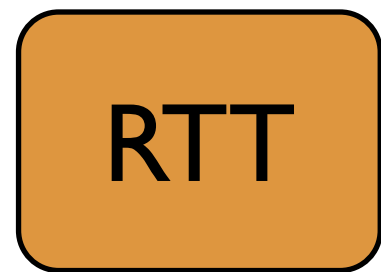
[2] A.Undrus et al. "Organisation and Management of ATLAS nightly builds", CHEP 2009



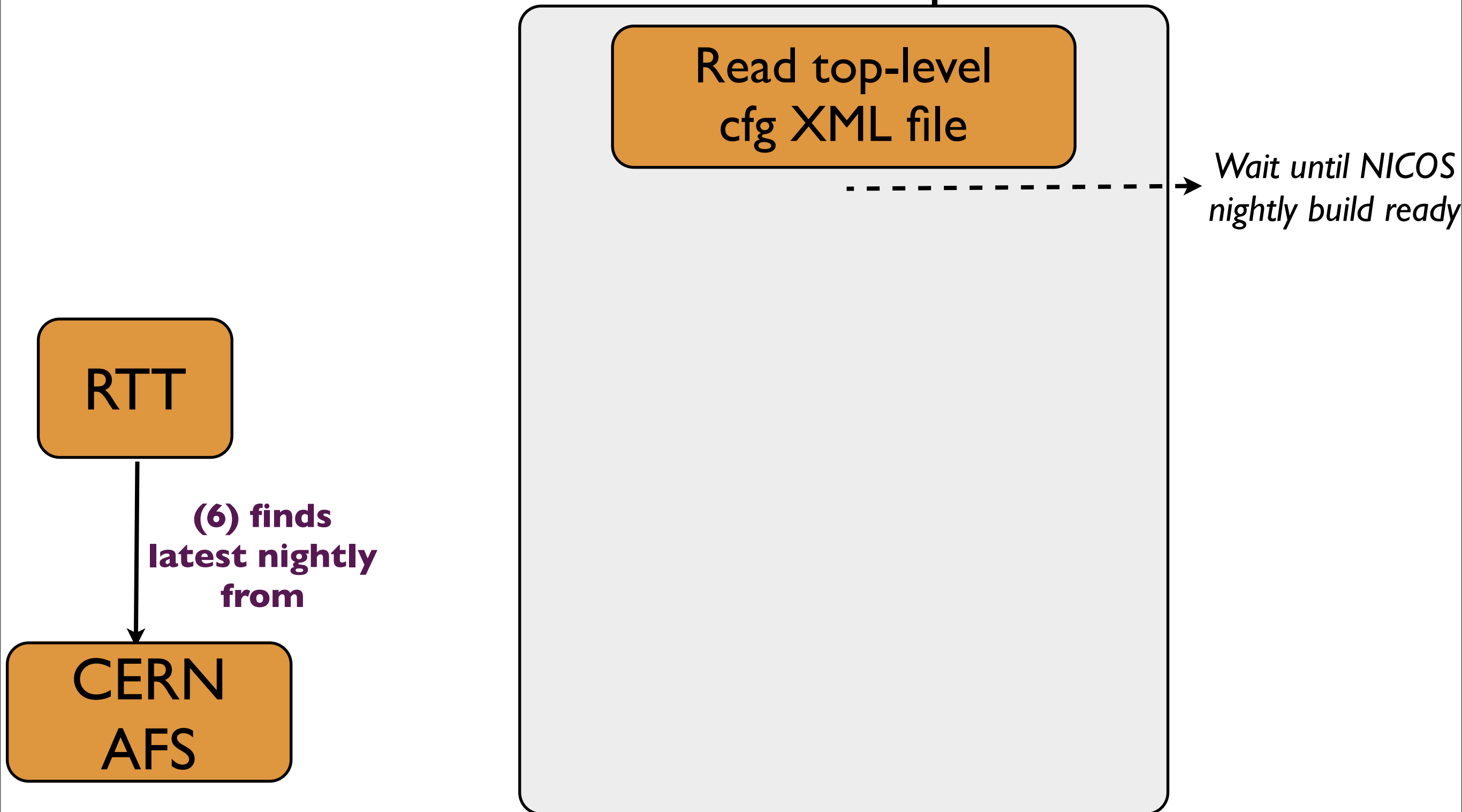
## RTT initialisation procedure



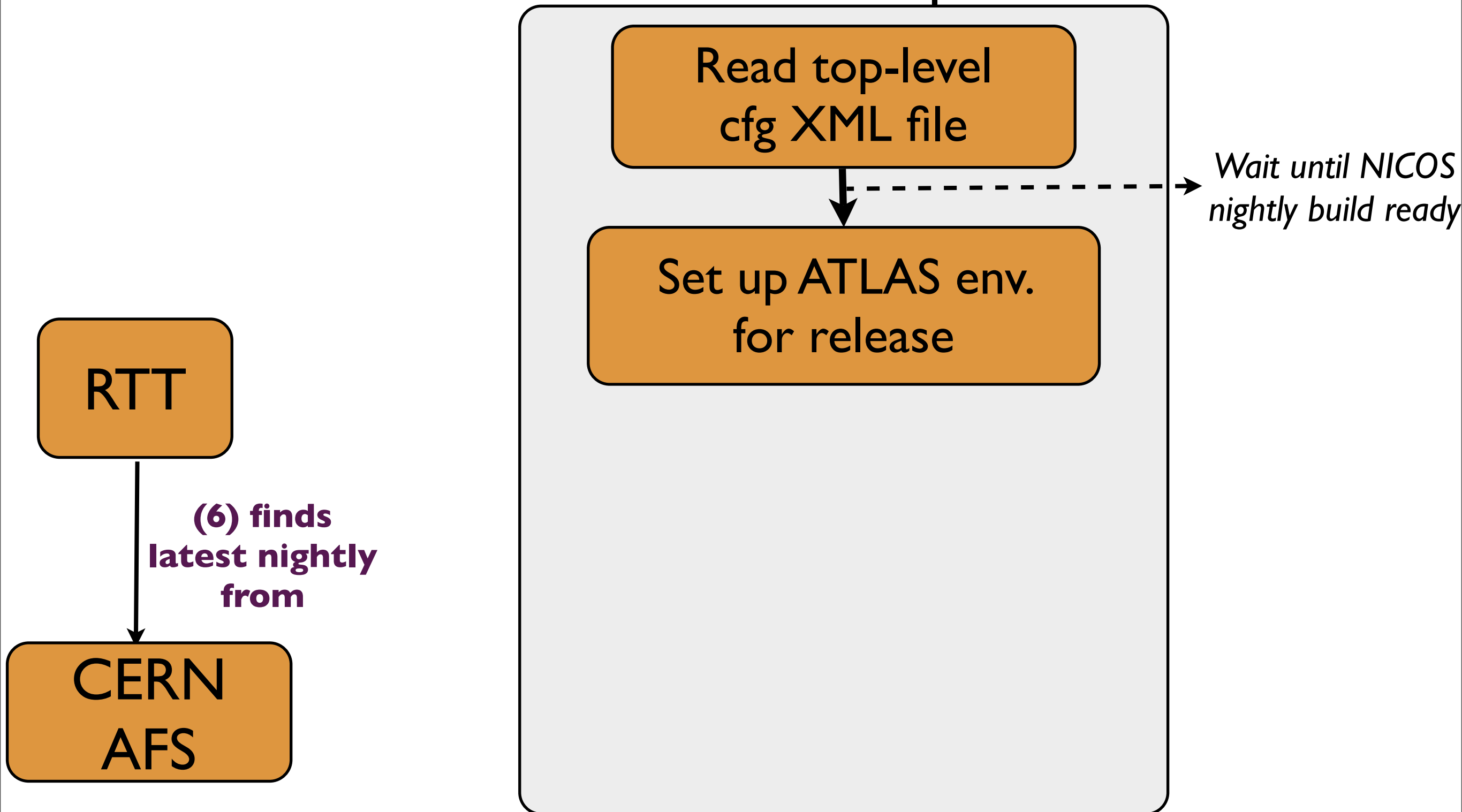
## RTT initialisation procedure



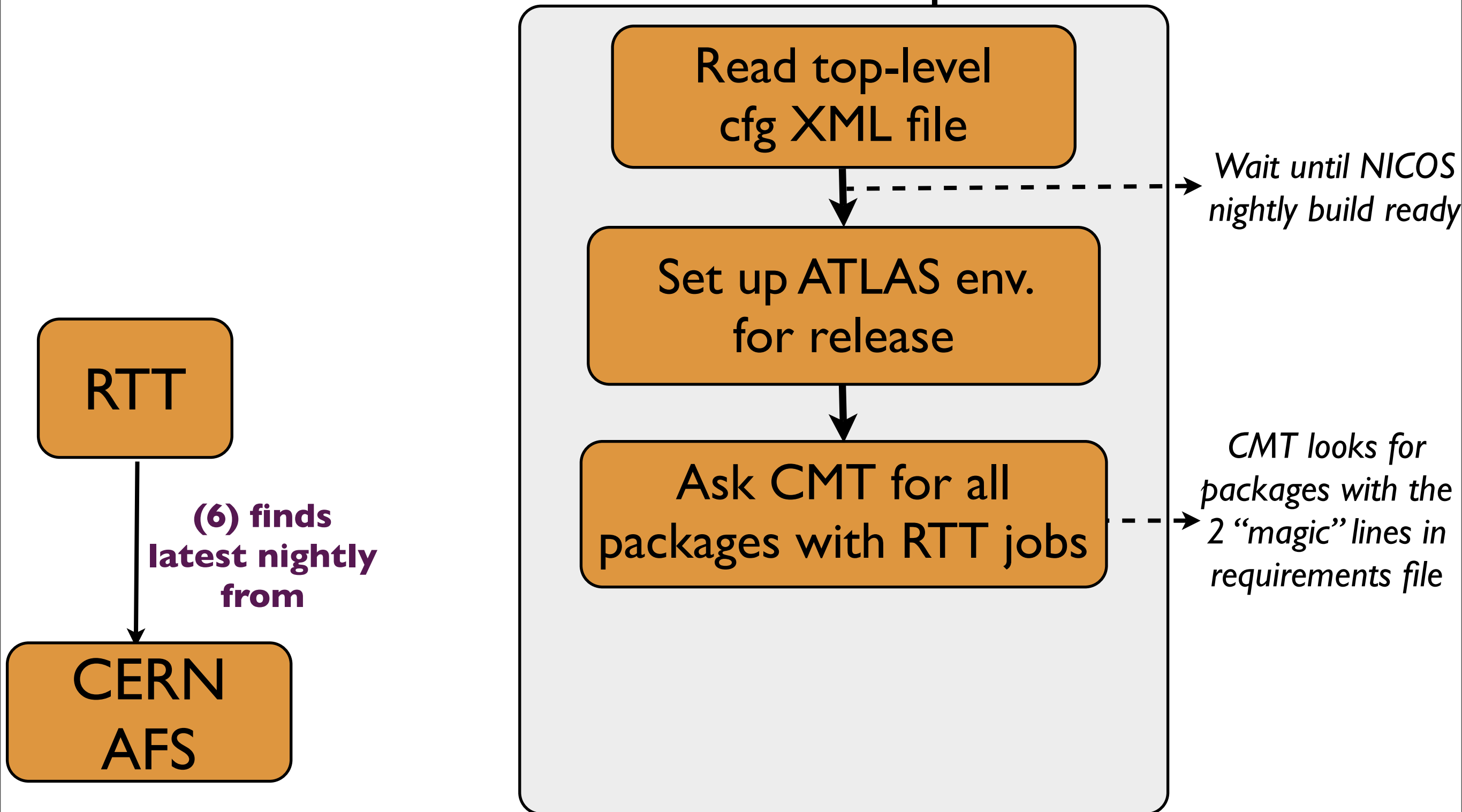
## RTT initialisation procedure



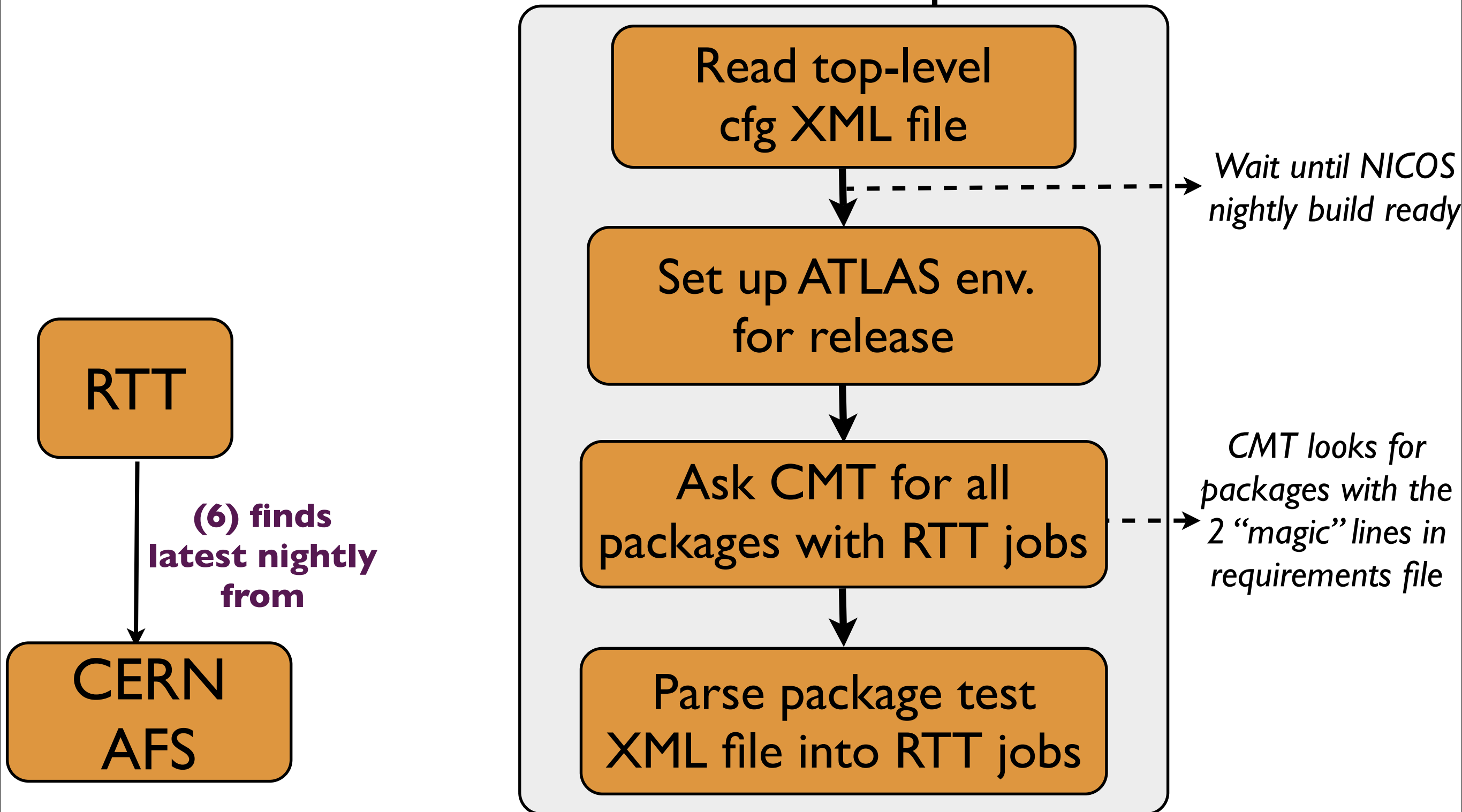
## RTT initialisation procedure



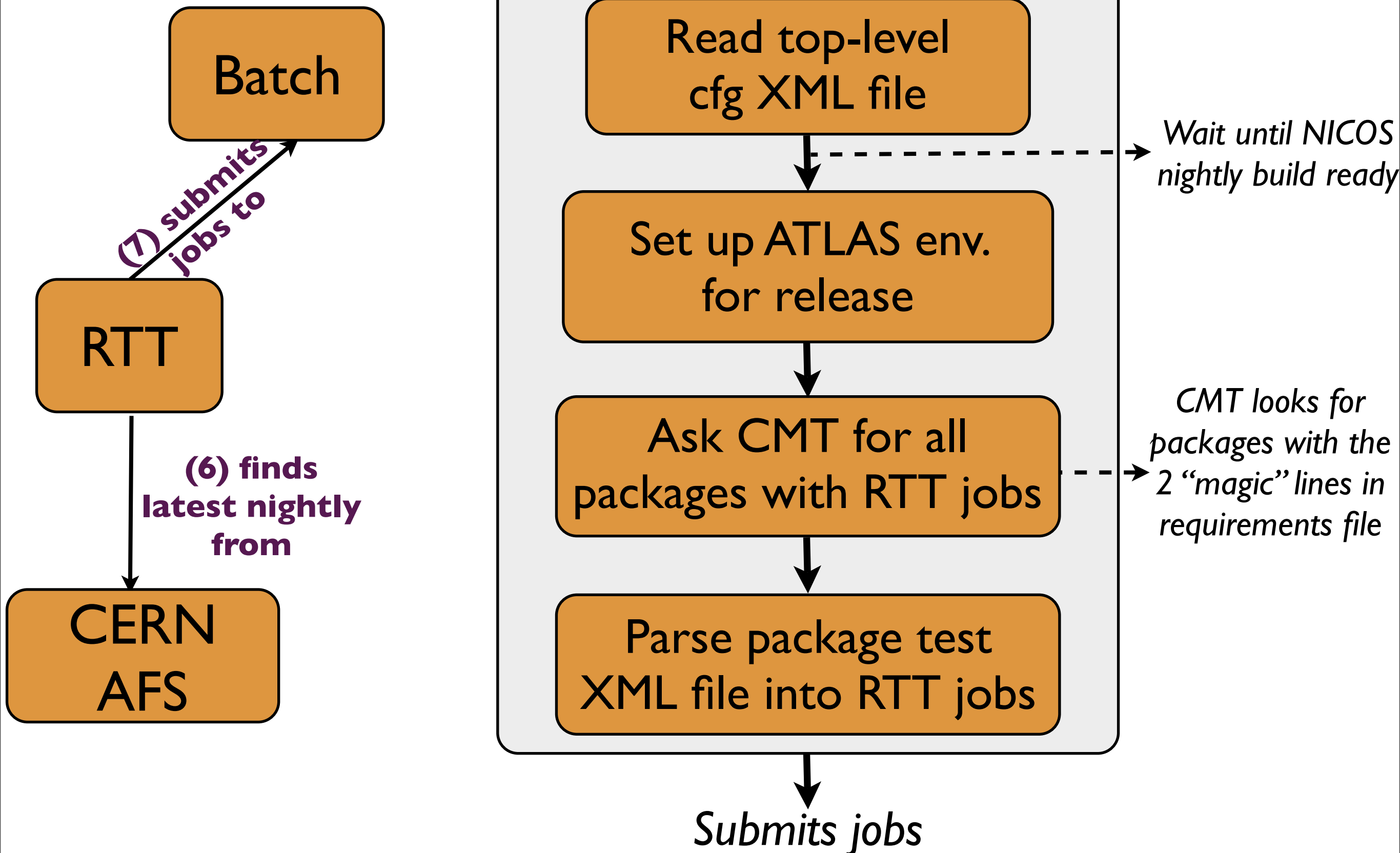
## RTT initialisation procedure



## RTT initialisation procedure



## RTT initialisation procedure



RTT currently runs at CERN:

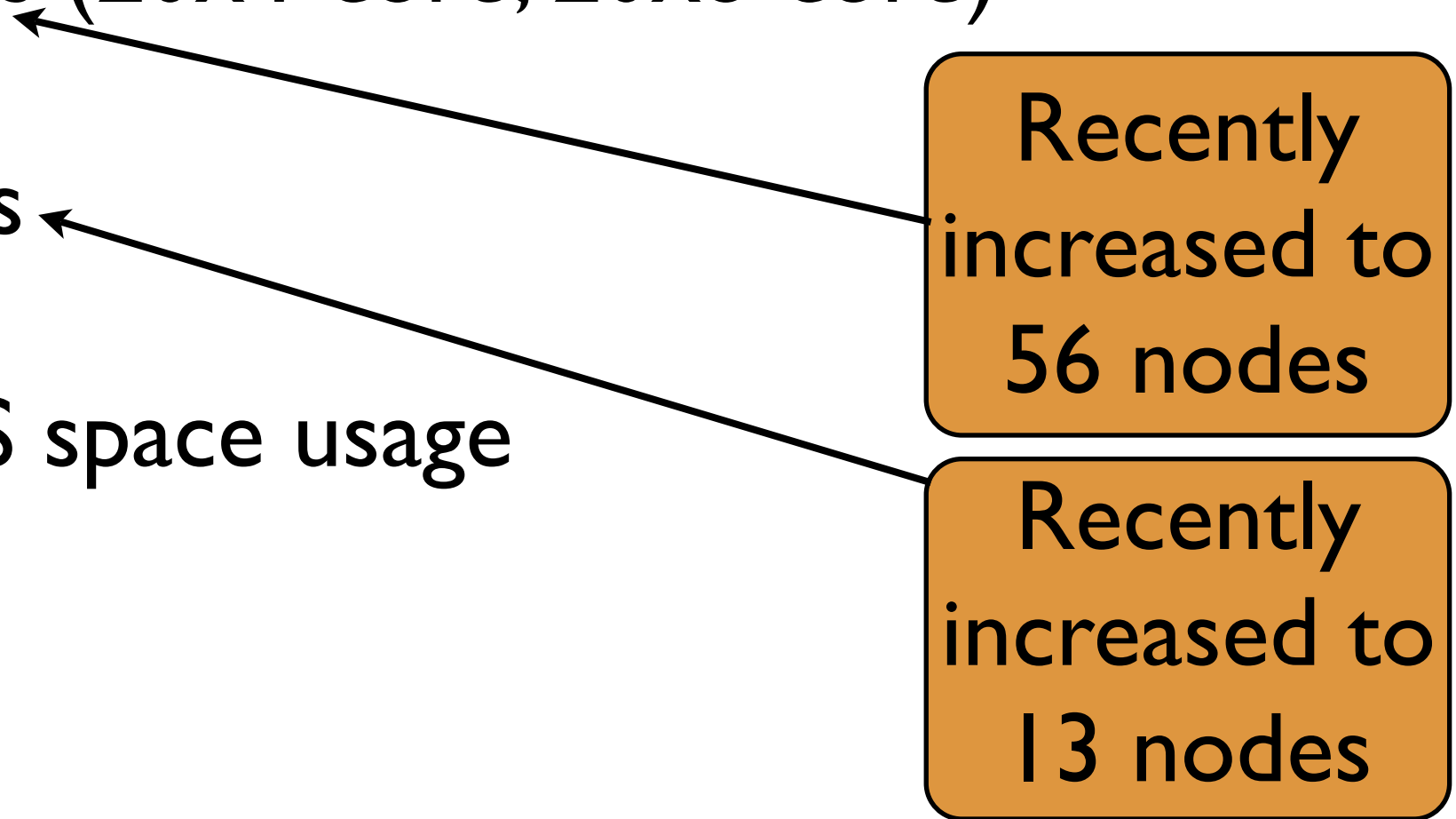
- 10 different ATLAS nightlies
- ~70 packages in largest nightly runs
- ~1300 RTT jobs/day total submitted to batch
- One platform: SLC4, gcc3.4



## Resources available to RTT:

- 3 private LSF batch queues
- 40 batch nodes (20x4-core, 20x8-core)
- 3 launch nodes
- ~1.4 TB of AFS space usage

Recently  
increased to  
56 nodes

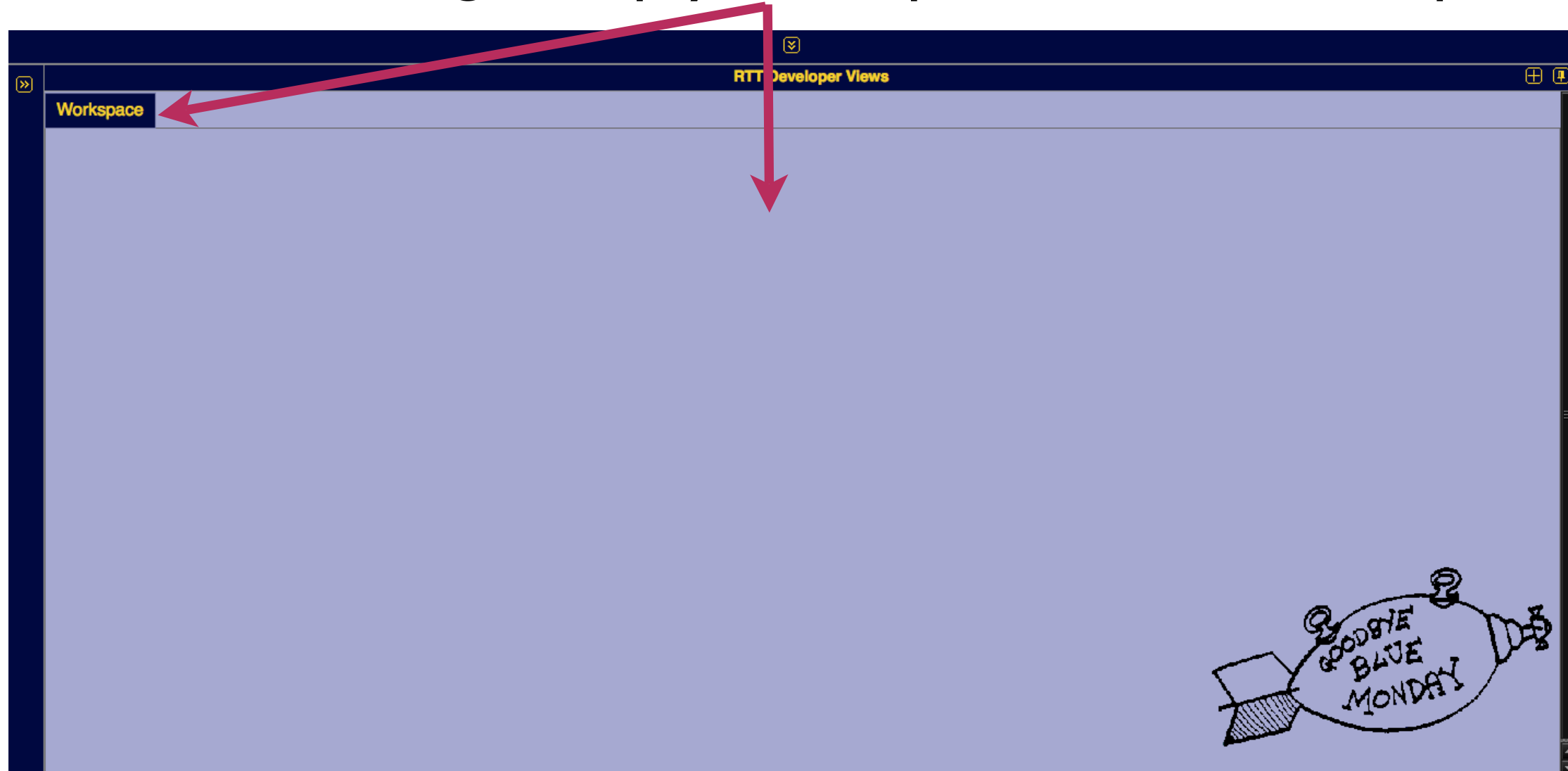


Recently  
increased to  
13 nodes

# The RTT

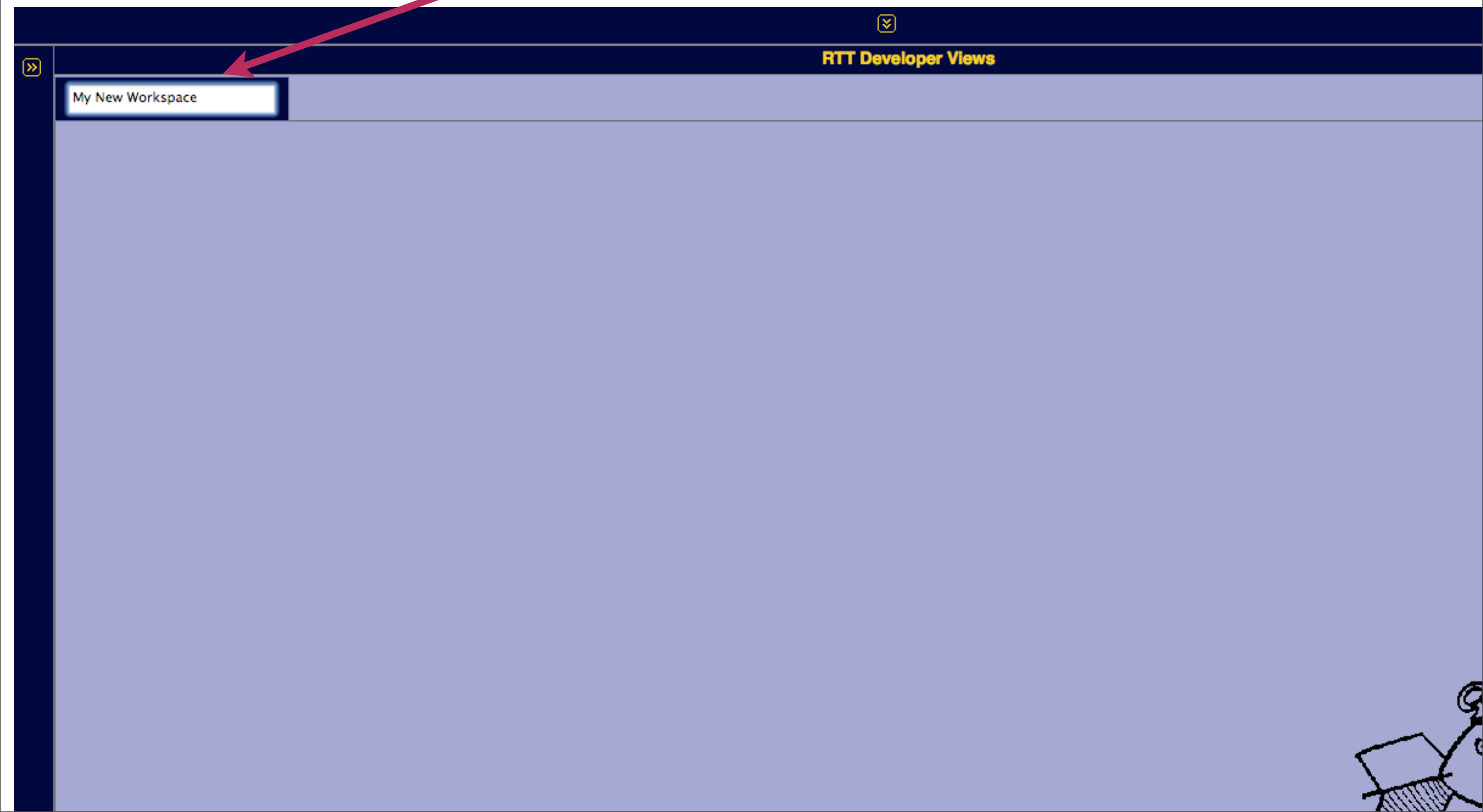
## Publishes

- Results published to <http://atlasrtt.cern.ch>
- First visit: a single empty workspace called “Workspace”

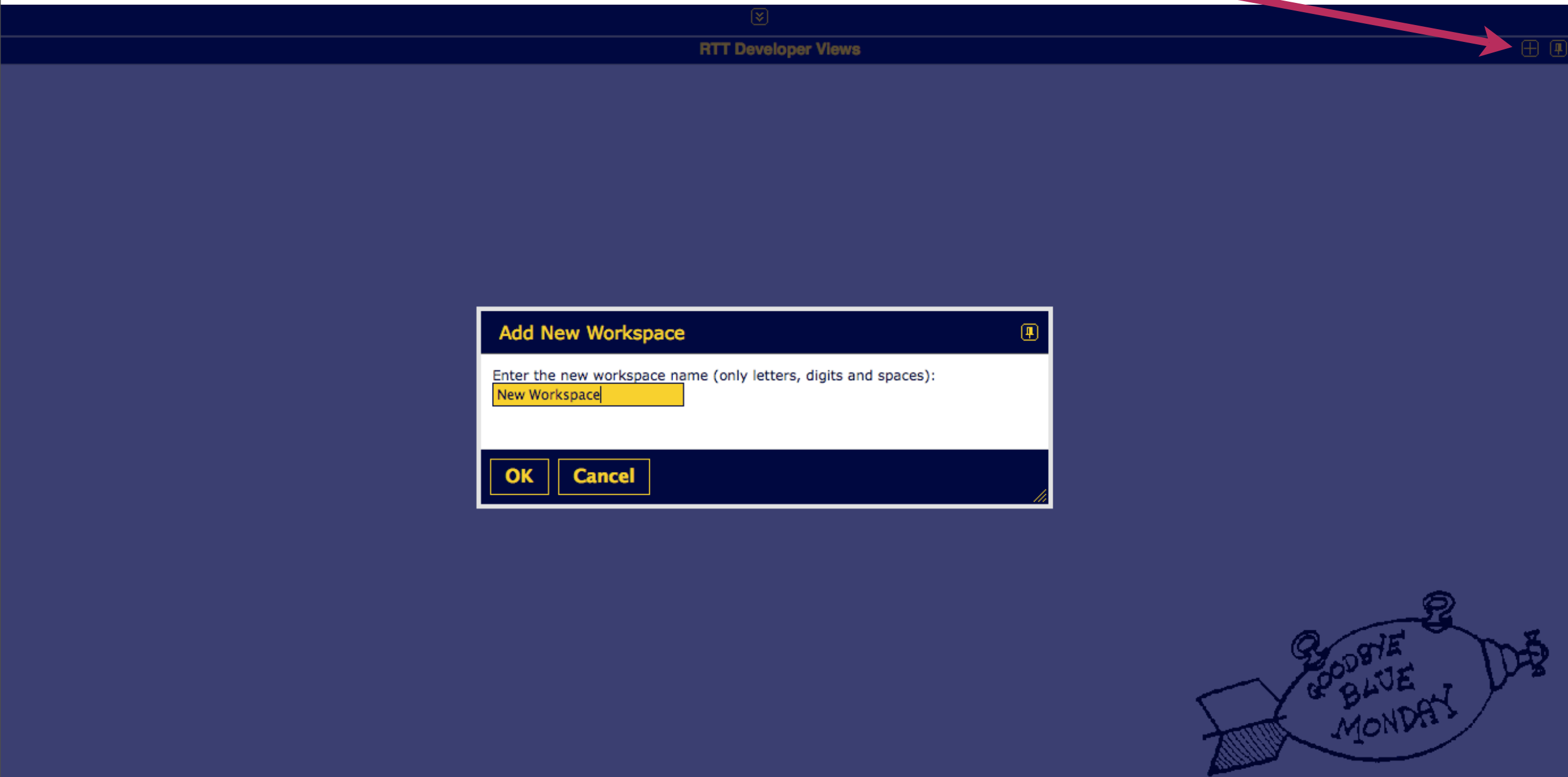


- A workspace will hold results for one or more packages
- A workspace can be customized by the developer
- “State” is persistent: subsequent visits to the site show last state (PHP sessions/cookies)

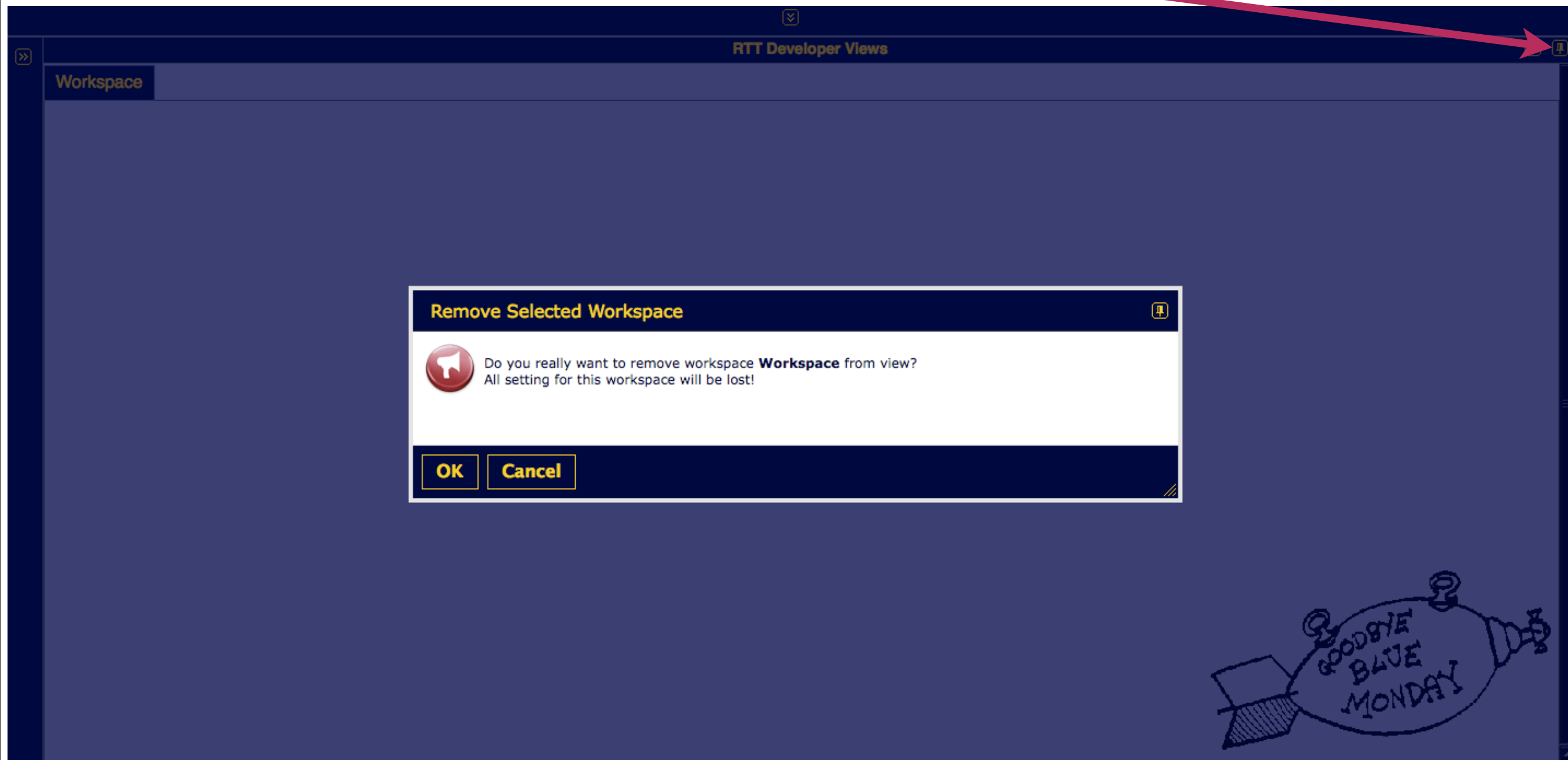
## Workspaces can be renamed....



New workspaces can be added....



Workspaces can be deleted....

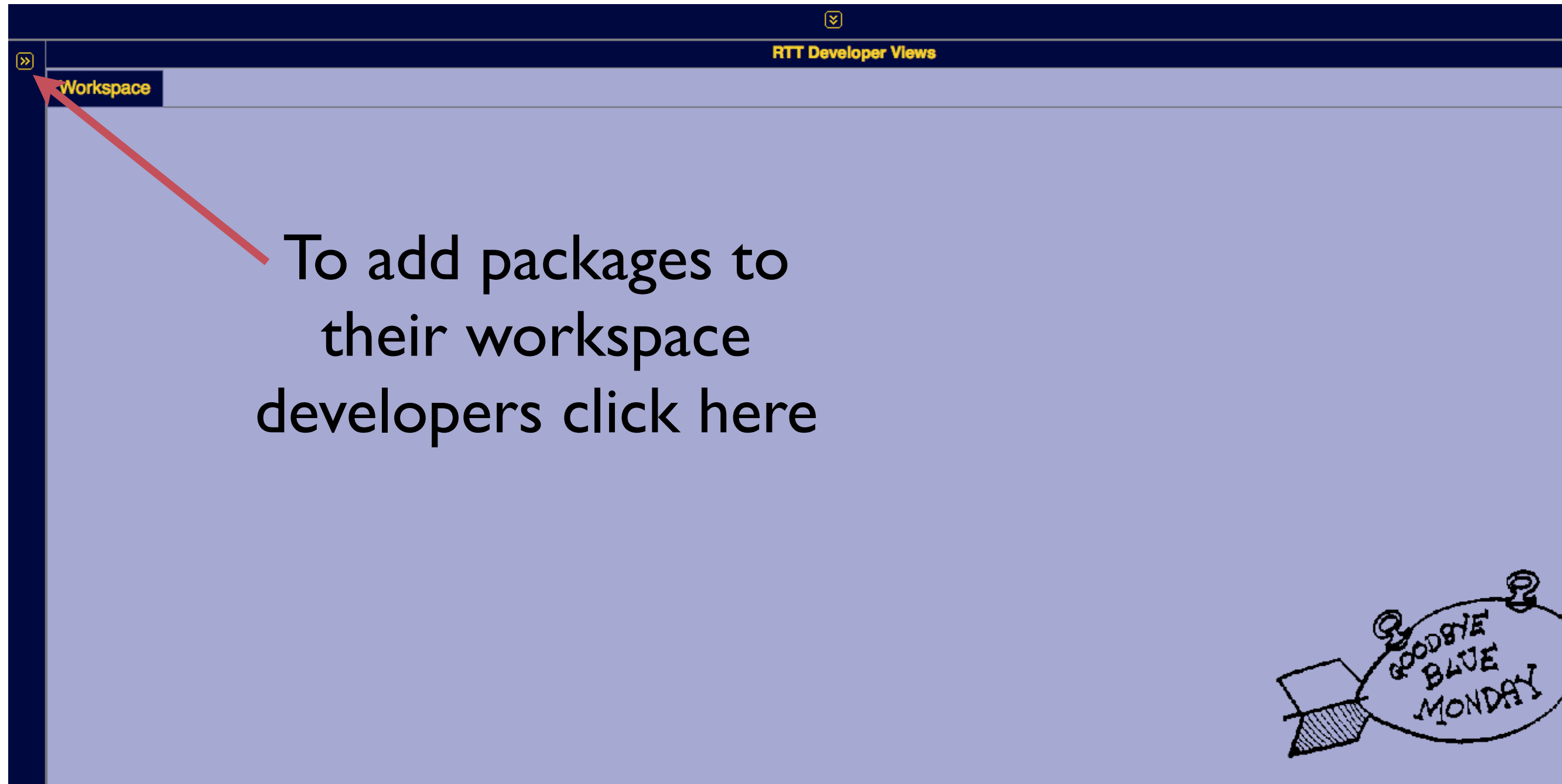


Within a workspace, package order can be changed by drag 'n' drop



# The RTT

➤ Publishes :: adding a package to a workspace

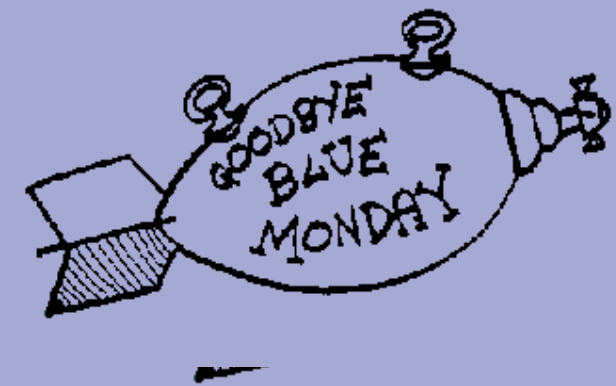
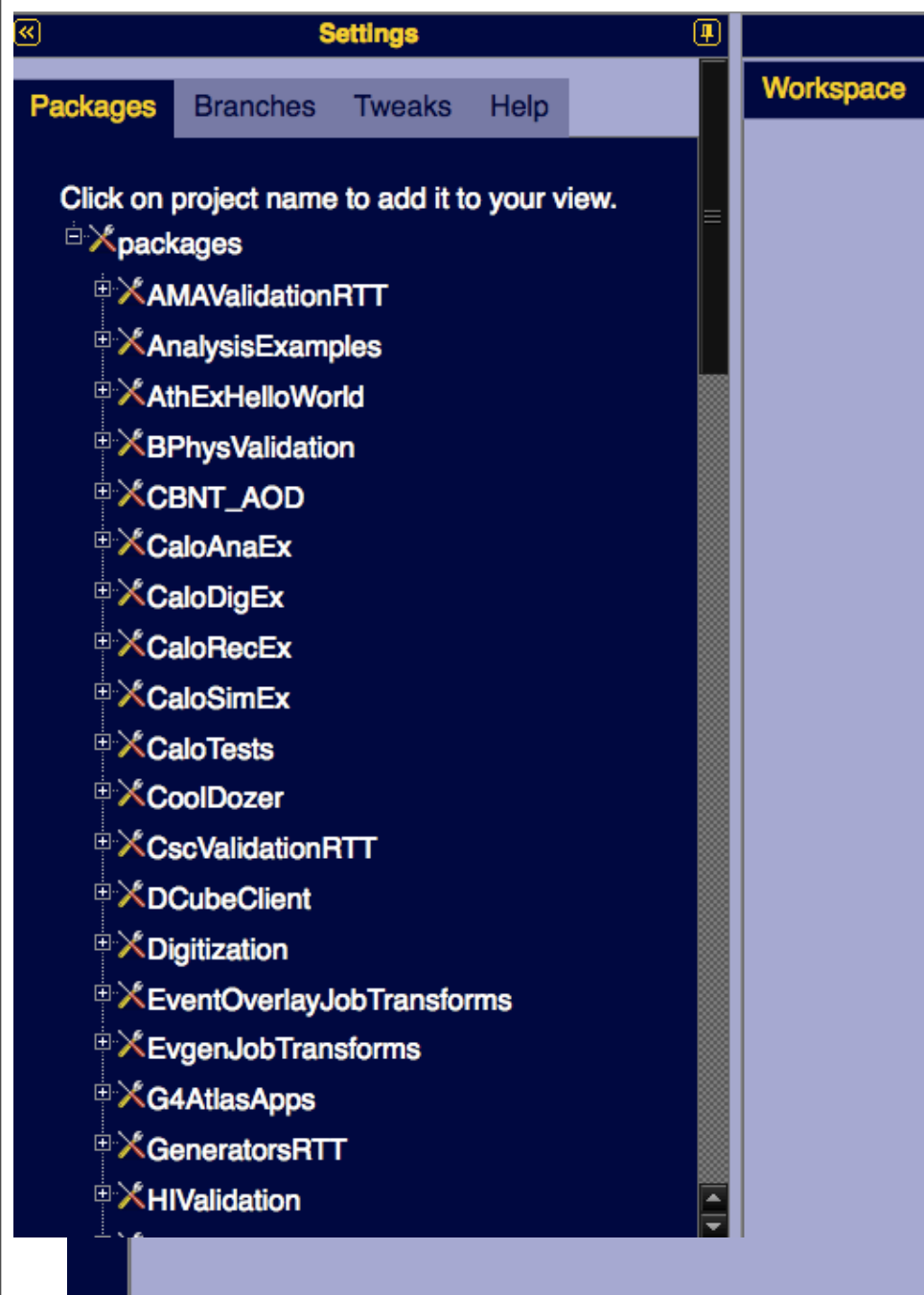




# The RTT

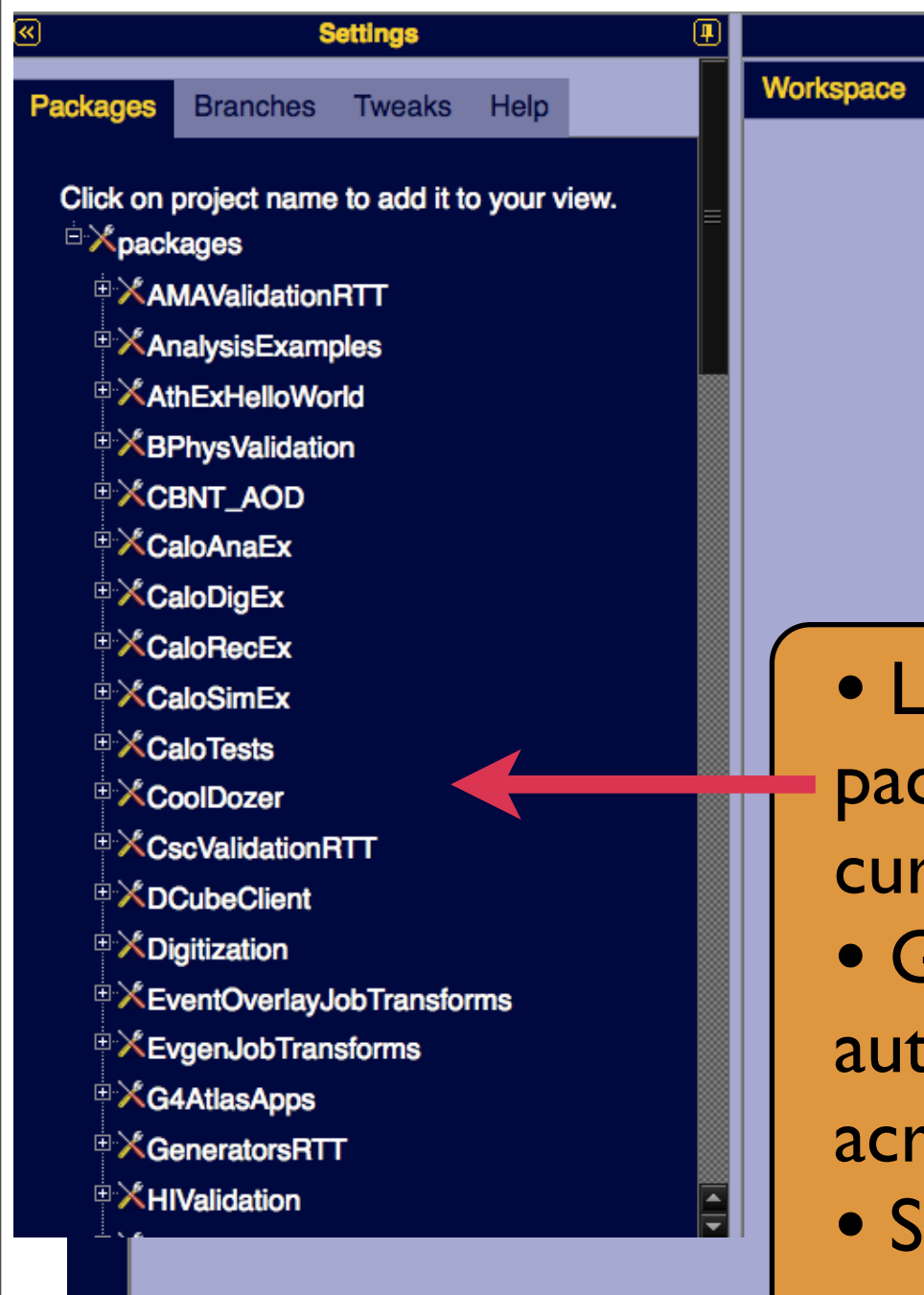
➤ Publishes :: adding a package to a workspace

- Settings pane slides open and pins itself
- Four tabs:
  - help
  - tweaks (change page styling)
  - packages
  - branches



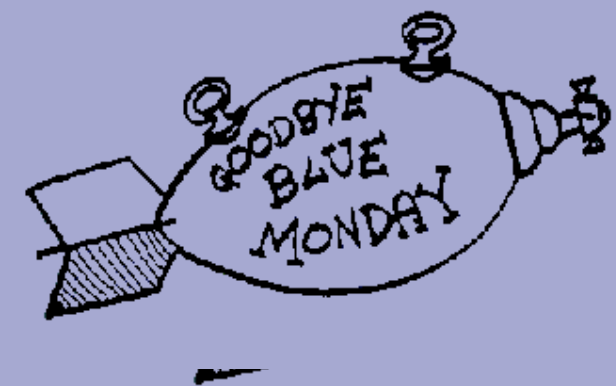
# The RTT

➤ Publishes :: adding a package to a workspace



- Settings pane slides open and pins itself
- Four tabs:
  - help
  - tweaks (change page styling)
  - packages
  - branches

- List of all known packages in all the current RTT runs.
- Generated automatically from the acron table
- Select a package, unfold a tree, and finally select the project heirarchy

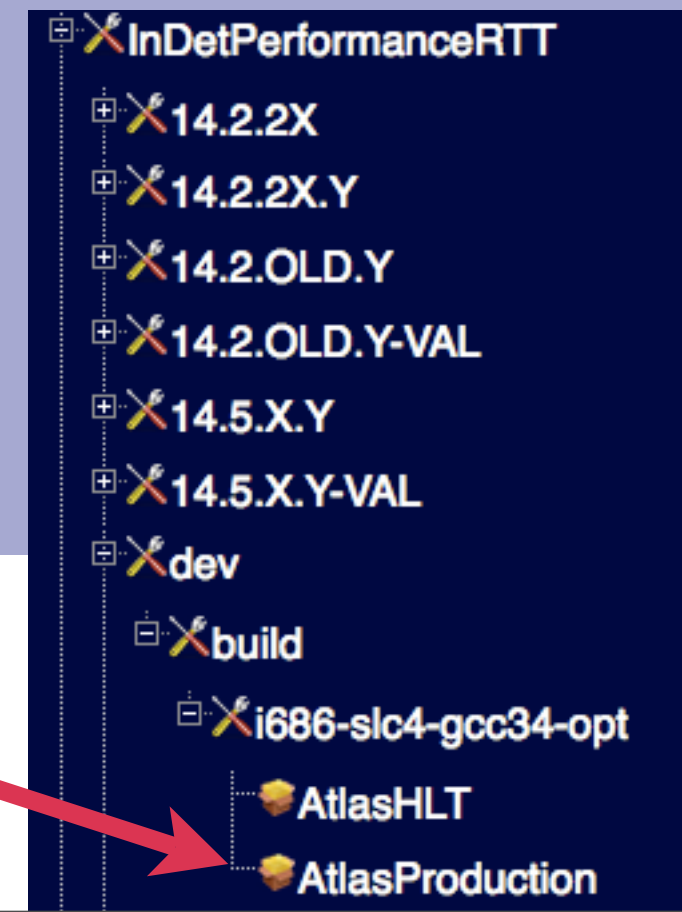
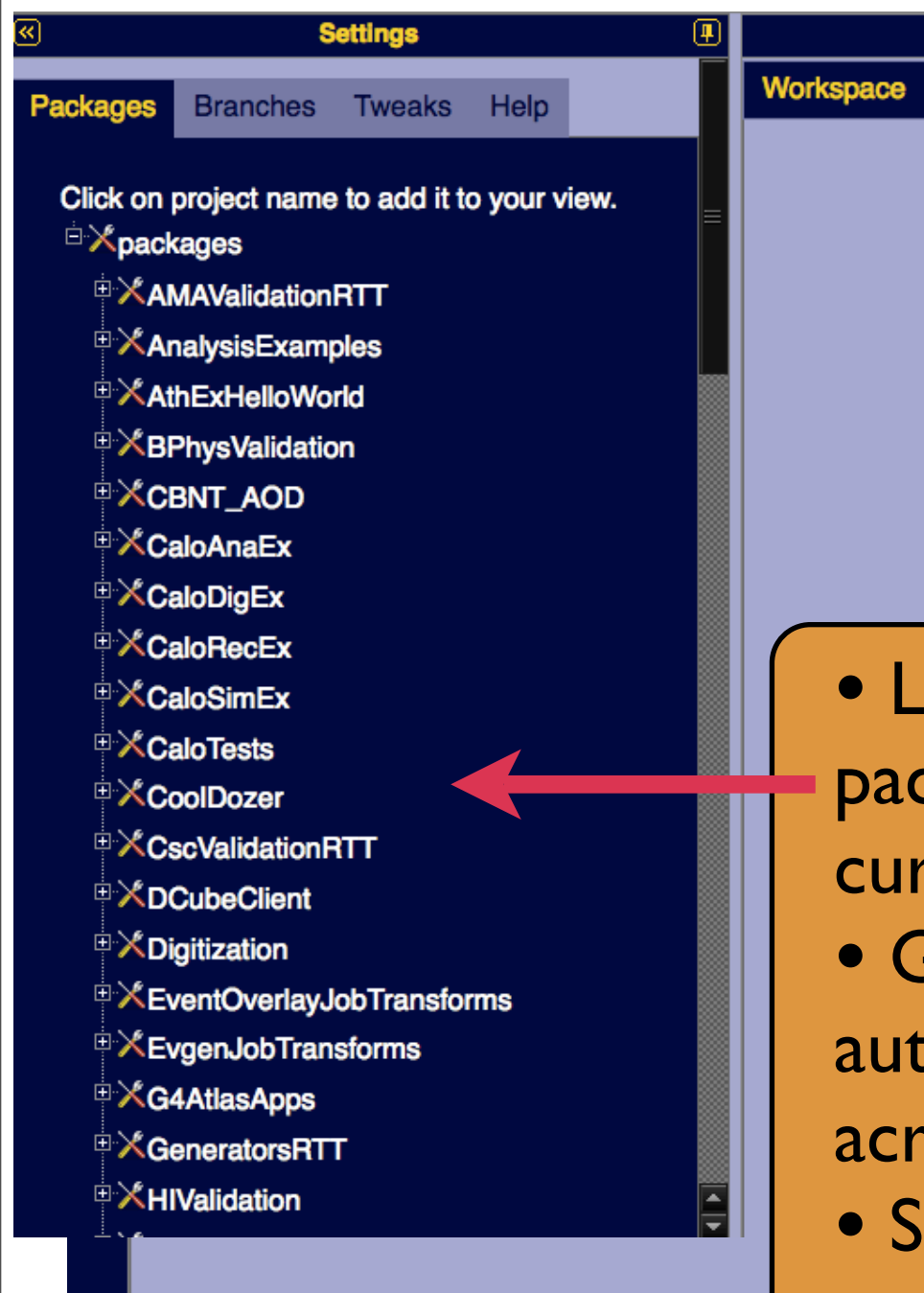


# The RTT

➤ Publishes :: adding a package to a workspace

- Settings pane slides open and pins itself
- Four tabs:
  - help
  - tweaks (change page styling)
  - packages
  - branches

- List of all known packages in all the current RTT runs.
- Generated automatically from the acron table
- Select a package, unfold a tree, and finally select the project heirarchy



# The RTT

➤ Publishes :: adding a package to a workspace

- Package for this run inserts itself into current workspace
- At-a-glance summary of the week's runs available

The screenshot displays the RTT interface with a workspace containing a package named 'InDetPerformanceRTT'. The package summary shows the following details:

- SOURCE CODE**: << ViewCVS >>  
CVS path: InnerDetector/InDetValidation/InDetPerformanceRTT  
CVS tag: InDetPerformanceRTT-00-00-49
- GRAPH**: << Atlas RTT Chain >>  
(experimental)
- LAST UPDATE**: Sun, 15th March 2009 06:39 PM
- JOBS**:

total:	6
done:	6
success:	6
failure:	0
closed down:	0
- TESTS**:

total:	0
success:	0
failure:	0

At the bottom, a list of jobs is shown, including 'InDetPerformanceRTT\_jobOptions [id: 278]', 'InDetPerformanceRTT\_jobOptions [id: 279]', and 'InDetPerformanceRTT\_jobOptions [id: 2811]'. A callout box on the left states: 'RTT jobs run for this package'.



# The RTT

➤ Publishes :: selecting a job within a package

## Select one of the RTT jobs: results unfold in situ

Summary info

Browsable Files and directories

✓ InDetPerformanceRTT\_jobOptions [id: 278]

**RESULTS**  
Batch status: done  
Job status: success  
Post-processing status: no tests

**HISTORY**  
queued: 09/03/15 11:34  
running: 09/03/15 15:06  
postProcessingQueued: 09/03/15 15:50  
postProcessingRunning: 09/03/15 15:51  
done: 09/03/15 16:55

**OVERVIEW**  
Job name: InDetPerformanceRTT\_jobOptions  
Job group: AthenaMu1IDPRTTKalman  
Job Id: 278  
Hash Id: 1259103062  
Ref release: 0.0.0

**CLASSIFICATION**  
Class: OfflineValidation  
Process: Reco  
Component: Det-InDet

**DOCUMENTATION**  
Description: Single Muons, 1 GeV, ATLAS-CSC-02-00-00, KalmanFitter  
Doc link: —

**FILES:**

InDetPerformanceRTT

DCube-InDetStandardPlots-SingleMu1000-Kalman-1

@ALL\_IDPerformanceRTT\_Plots.html

ALL\_IDPerformanceRTT\_Plots\_Dynamic.html

ERROR.log

IDPerformanceRTT\_dynamic\_pagemaker.log

@IDPerformanceRTT\_dynamic\_pagemaker.py

@InDetPerformanceRTT\_TestConfiguration.xml

InDetPerformanceRTT\_jobOptions278\_log

InDetStandardPlots.root

JobCompleteMarker.txt

PerfMonDozer.log

PoolFileCatalog.xml

ROOToutput.log

RTTtests.db

badmatchrate\_eta.png

badmatchrate\_pt.png

cmtShowUses.log

effplot0.png

effplot1.png

effplot2.png

env.log

fakerate\_eta.png

fakerate\_pt.png

hiteff\_0.png

# The RTT

➤ Publishes :: emailing a package

1) Right-click, copy URL for this package

**InDetPerformanceRTT (dev — build — i686-slc4-gcc34-opt — AtlasProduction)**

rel\_2 rel\_3 rel\_4 rel\_5 rel\_6

**SOURCE CODE** << ViewCVS >>  
CVS path: InnerDetector/InDetValidation/InDetPerformanceRTT  
CVS tag: InDetPerformanceRTT-00-00-49

**GRAPH (experimental)** << Atlas RTT Chain >>  
**LAST UPDATE** Wed, 18th March 2009 05:08 PM

**JOBS**

total:	6
done:	6
success:	6
failure:	0
closed down:	0

**TESTS**

total:	0
success:	0
failure:	0

InDetPerformanceRTT\_jobOptions [id: 278]

2) Email URL to friend

3) Friend clicks on URL in mail

4) Default browser opens: workspace+package installed

# Summary

- The RTT is a framework used to test ATLAS offline software
- It is a major component of the ATLAS software validation effort
- It runs daily in production at CERN...
- ...but can also be downloaded and run locally by developers
- In current production at CERN:
  - ▶ Processes 10 ATLAS nightly builds
  - ▶ Submits jobs from 3 launch nodes...
  - ▶ ...to a 40-node cluster with 3 LSF batch queues
  - ▶ Results are published to <http://atlasrtt.cern.ch>

## RTT User Guide

<http://www.hep.ucl.ac.uk/AtlasTesting/RTTUserGuide/RTTUserGuide.html>

## RTT Production Results web pages

<http://atlasrtt.cern.ch>

## RTT Developers

Peter Sherwood/Brinick Simmons/Alexander Richards ([rtt@hep.ucl.ac.uk](mailto:rtt@hep.ucl.ac.uk))

## RTT Web Developer

Krzysztof Ciba ([ciba@galaxy.uci.agh.edu.pl](mailto:ciba@galaxy.uci.agh.edu.pl))