

# Rationalizing the software development cycle at Colruyt

# Agenda

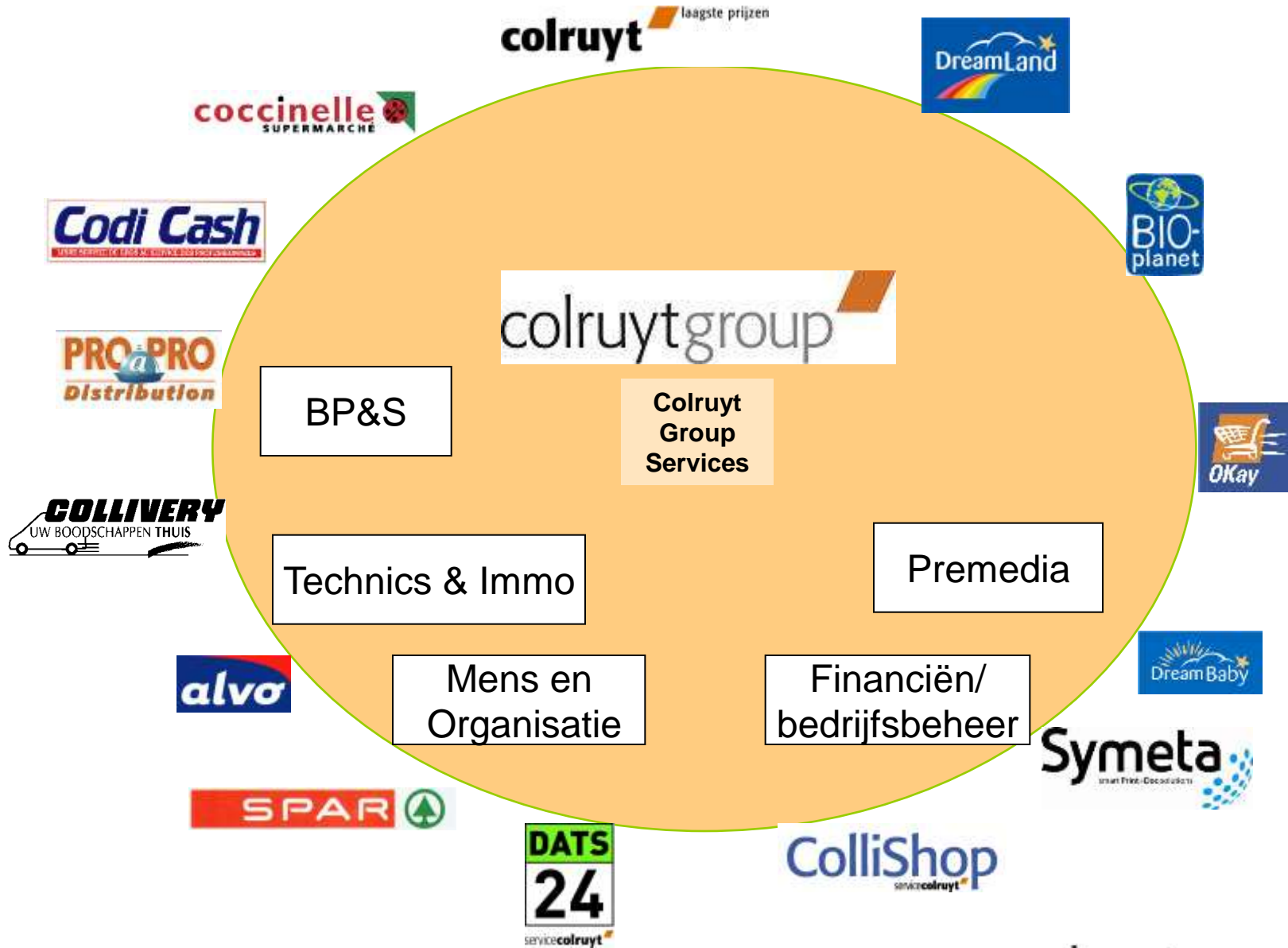
→ Introduction

**Project Context**

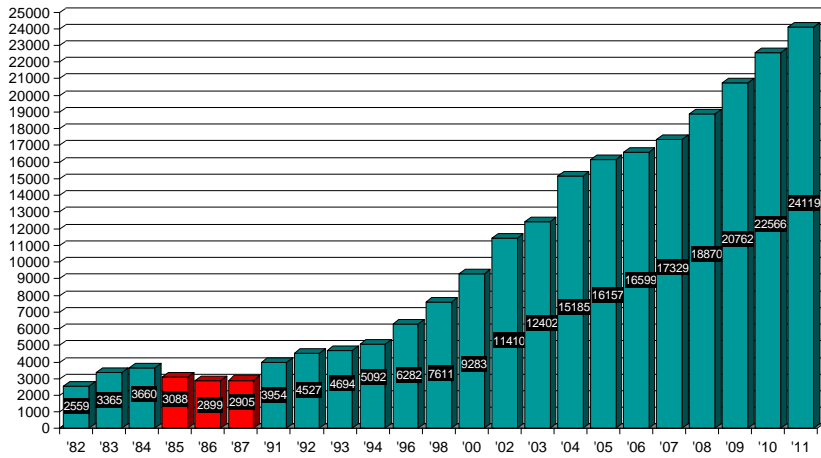
**High Level Solution View**

**Implementation**

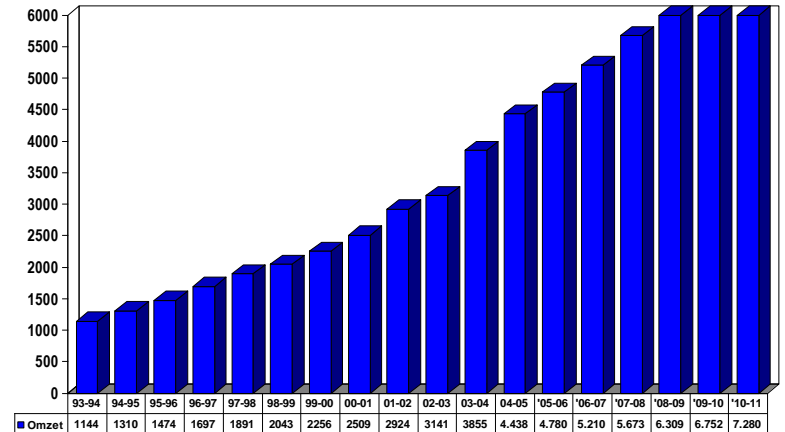
**Q&A**



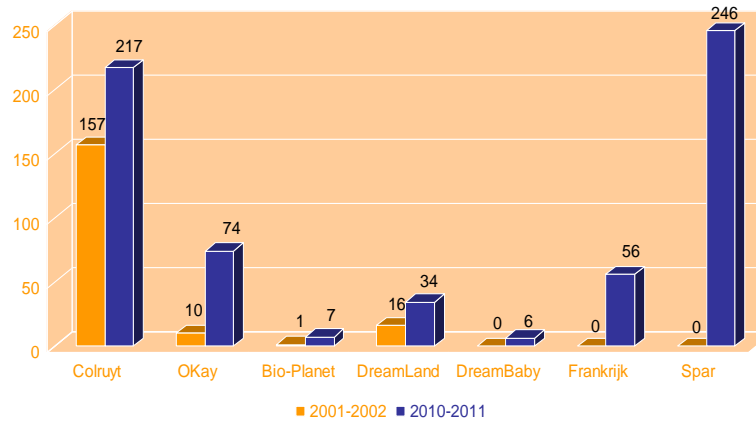
# Growth



Personeelsbestand



Omzet



Verkooppunten

# Diversification in activities

## Distribution



## Production & Packaging



## Technics&Immo, Garage, Architecting, Energy



## IT, Finance, HR, Photostudio, Marketing, ...



# Evolutions in IT





*SIMPLY EXPLAINED - PART 37:  
AGILITY*

# Challenges

- Cope with increasing rate of change of both business and technology
- Manage the upscaling on all fronts
- Manage and control diversification of the technology landscape

## **OUR GOAL**

**Optimize the development environment and processes to ensure stability and throughput of IT**

# Agenda

**Introduction**

**→ Project Context**

**High Level Solution View**

**Implementation**

**Q&A**



# Project Context

- At the start one all encompassing program
  - Organizational processes
    - Also a business component within IT
    - Introduction of business, enterprise and solution architecture
    - Introduction of Centers of Excellence
    - Introduction of new functions and roles
  - Creation and rollout of a new Software Delivery Lifecycle Process
  - Reassessment of the IT landscape and development environment
    - Change in mindset : only reinvent the wheel on the parts that are crucial to distinguish Colruyt from its competition
      - More “bought” technologies and packages
- Then broken up in parts with reduced scope. A lot of links were cut ; “the dots will be joined later”
- One of these parts is our project to introduce a new software development environment for non-legacy software (develop, build, test, deploy)

# Project Context

- With this in mind it is important that our development environment
  - Immediately delivers added value to its stakeholders
  - Works in a multi-site environment
  - Supports different development processes
  - Is **flexible** and widely employable
    - Easily support different technologies, changes on infrastructural level, cross-technology orchestration, ...
  - Scores high on **integration**
    - To be able to connect some dots later on
- Our current environment scores high on efficiency and control but not on flexibility and integration
  - Home grown tools in addition to RAD and PVCS version manager
- Starting point for our solution :  
**a “basic” solution that can then evolve in lign with decisions still to be taken in other tracks**

# Approach

- Requirements definition
- Selection track aligned with our IT principles
- Proof of concept with RTC
- Work out a high level solution based upon RTC
  - BF and RAM became part of the solution
- Architectural track in close collaboration with IBM Solution Architect
  - Focus : only java software development but keeping all the “dots” in mind
- Implementation Tracks
- Migration Tracks
  - RAM as one-shot
  - RTC and BF as team-by-team
- Trainings

# Agenda

**Introduction**

**Project Context**

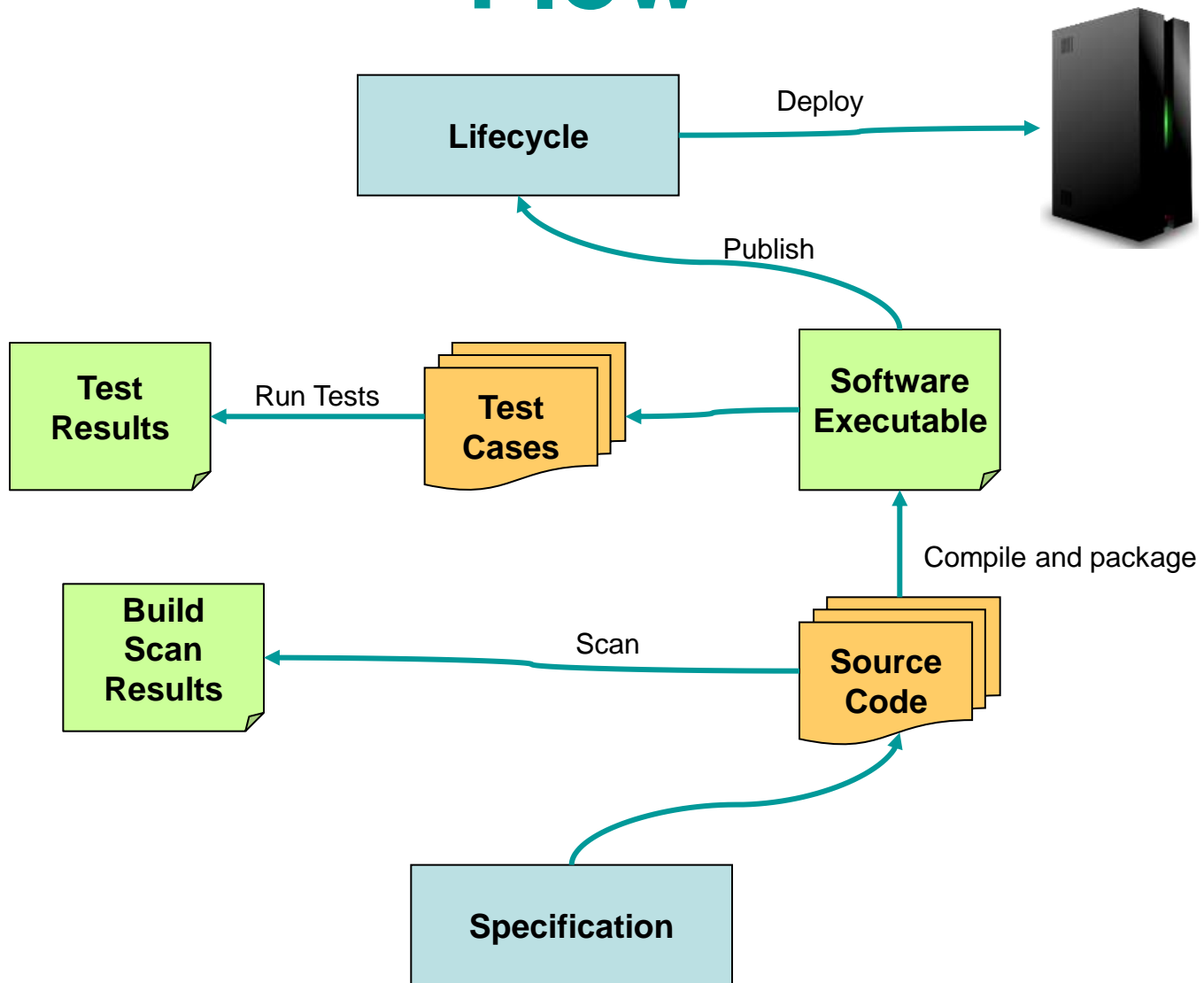
**→ High Level Solution**

**Implementation**

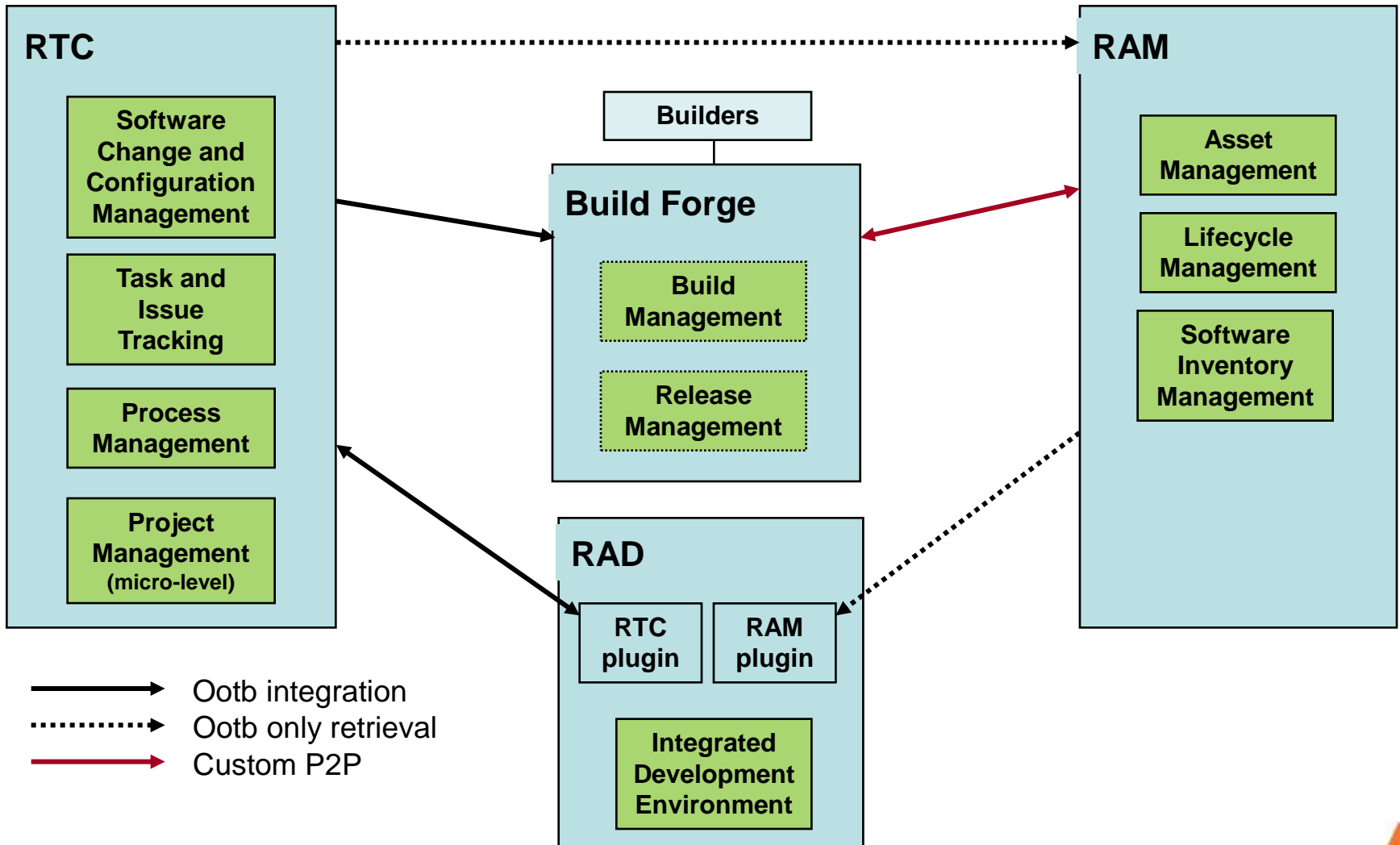
**Q&A**



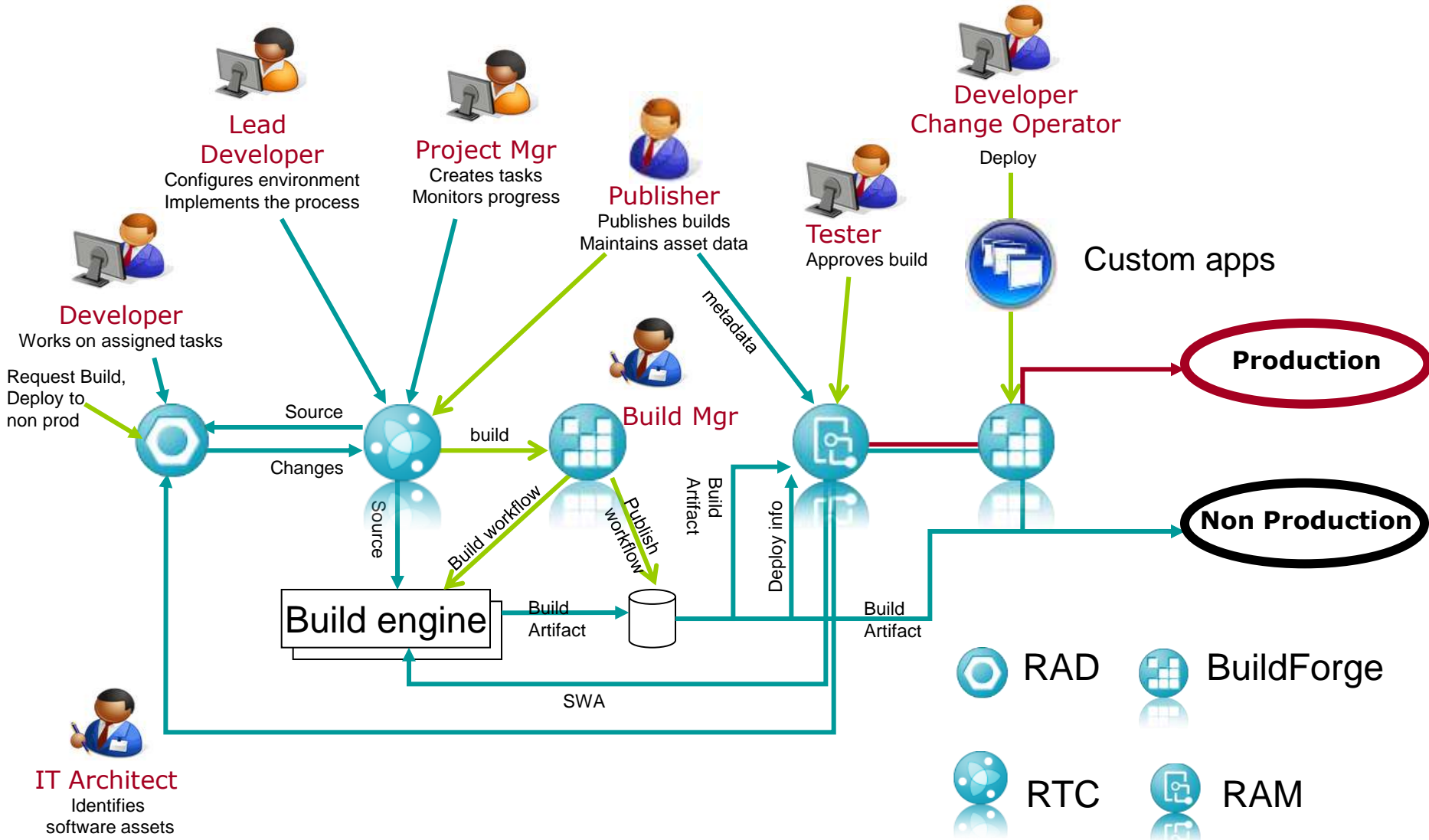
# Flow



# Components



# Architectural Overview



# Agenda

**Introduction**

**Project Context**

**High Level Solution View**

**→ Implementation**

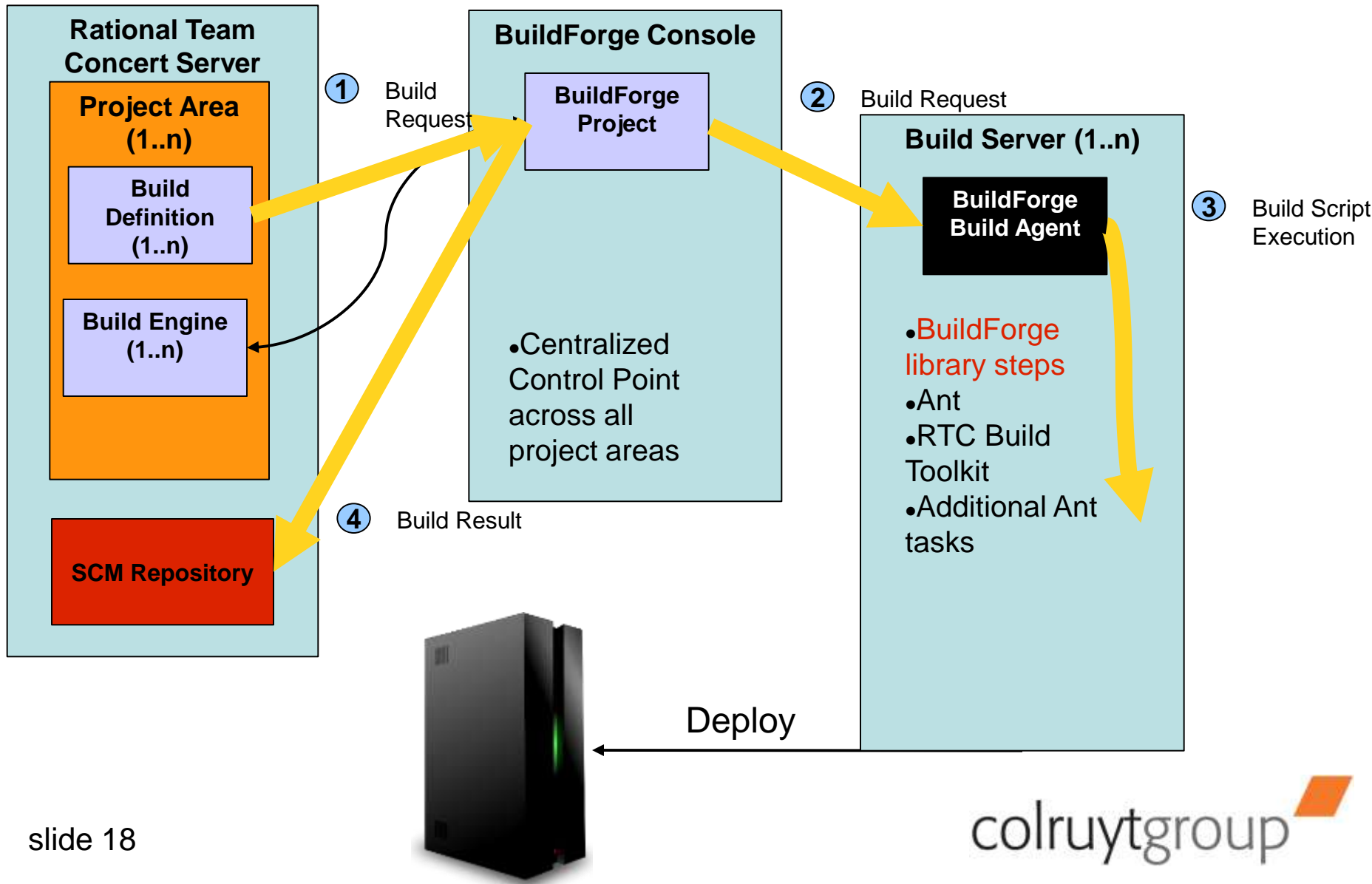
**Q&A**



# Why we use RTC together with BF

- To accomplish our principle : **build once – promote many**
- Apply company-wide build processes with steps that are easily reusable
- Global logging, monitoring
- Continuous integration, scalability
- Note : we use BF as a tool working “behind the screens”.

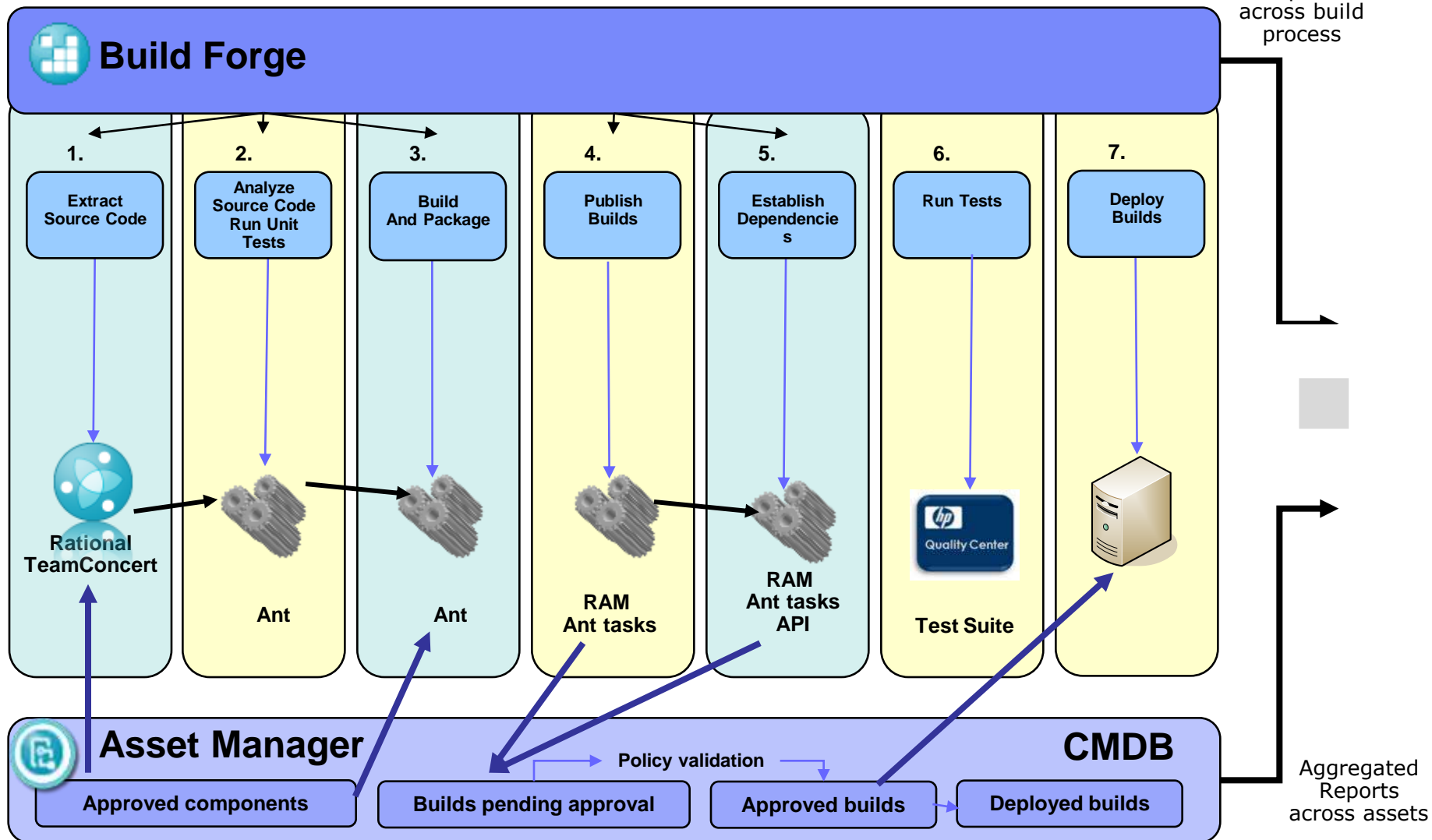
# Using RTC and BF



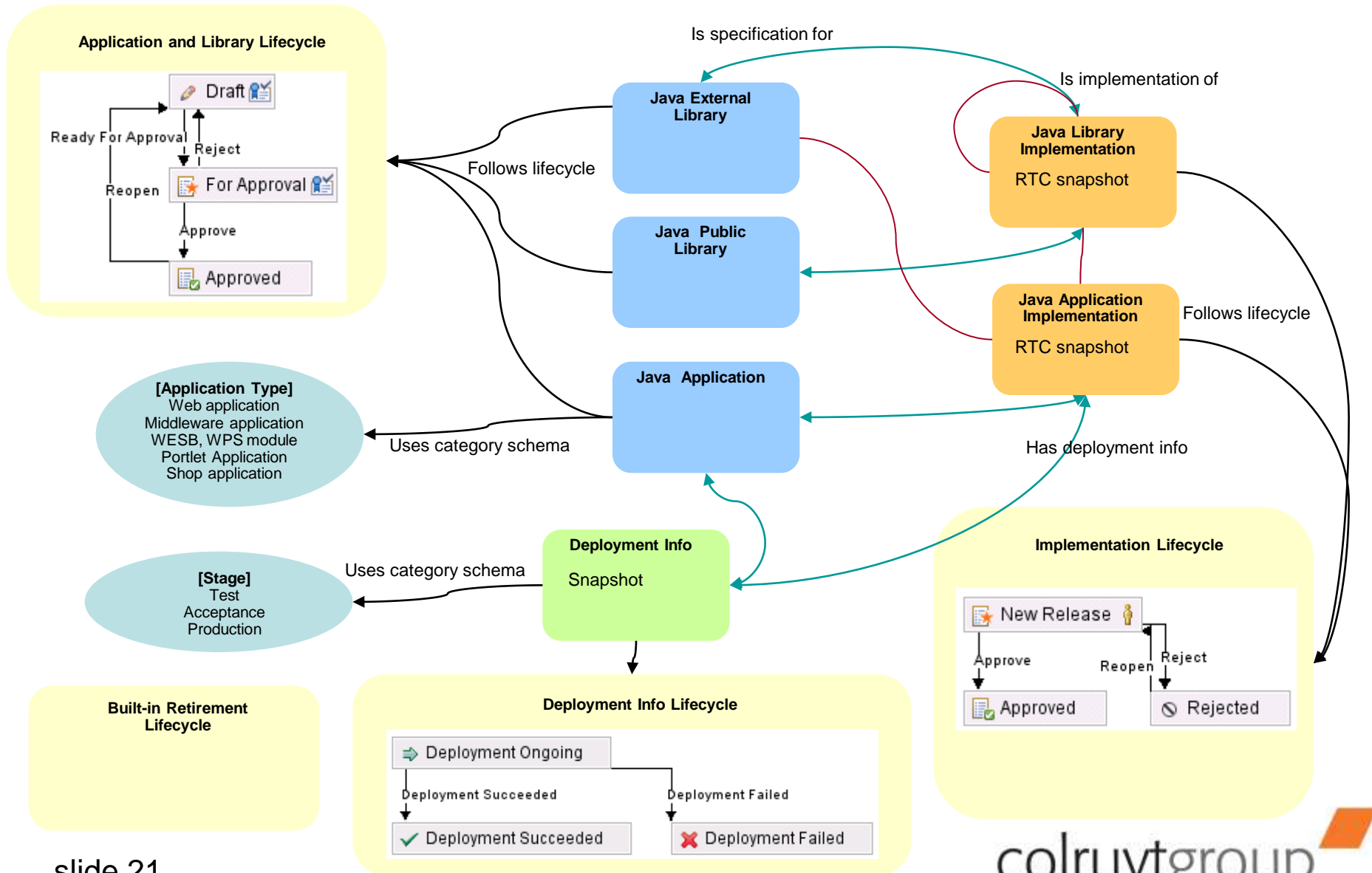
# Why we added RAM to the mix

- As a software library
- As a broker between asset providers and consumers
  - Only use approved assets
- Tracking dependencies between software assets, platforms, external components, runtime environments, ...
  - Through BuildForge build steps
- ALM functionality
  - Eg approval processes, retiring components

# Adding RAM to the mix



# RAM Configuration



# Considerations

- Important to get familiar with the product landscape of the vendor
- A common “language” is important
  - All parties, internal and external, need to understand each other thoroughly
  - Work together on-site
- Generic tools
  - Investment needed to draw and implement a design that fits your needs : it will take time !
  - Trade-off flexibility vs “ready-to-eat”

# Agenda

**Introduction**

**Project Context**

**High Level Solution View**

**Implementation**

**→ Q&A**

