



IBM Software Group

IBM Unified Software Configuration Management on z/OS

IBM SWG Zhao Jun

Rational software



ON DEMAND BUSINESS™

© 2005 IBM Corporation

Agenda

- z/OS Software Configuration Management Challenges

- IBM Solution
 - ▶ Solution Overview
 - ▶ Architecture and Deployment Model
 - ▶ Backend Technical Detail

- Case Study and Implementation Discussion



z/OS Software Configuration Management Challenges

- No change management or change management separate from z/OS asset
 - ▶ No automatic connection between change request and modified asset
 - ▶ Need audit trail for change activity (who, what, when, and where)

- No unified configuration management for composite application
 - ▶ Back-end z/OS and front-end distributed parts are controlled by different system
 - ▶ No base line for entire application and No Comprehensive inventory of all software assets

- High level parallel development depends on manual work

- Can not implement unified development process due to z/OS uniqueness



Agenda

- z/OS Software Configuration Management Challenges

- IBM Solution
 - ▶ Solution Overview
 - ▶ Architecture and Deployment Model
 - ▶ Backend Technical Detail

- Case Study and Implementation Discussion



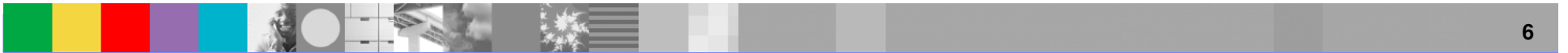
Solution Overview

- Provides single point of control, single repository of knowledge, single artifact repository, single set of user interfaces
- Provides consistent process paradigm across platforms
- Extends the power of ClearCase, ClearQuest, & Build Forge to z/OS development
 - ▶ Parallel development; powerful workflow; powerful build capabilities
- Extends the power of the Software Delivery Platform to z/OS development



IBM Rational SCM Solution Value

- Dramatically reduce TCO by moving as many as possible activities to distributed side
- Unified SCM tools, process and SCM team
- Improves developer productivity by fully supporting parallel development
- Improves availability of z/OS applications by automation of build & deploy tasks.
 - ▶ Stronger parallel development and release management
- Provides compliance and governance by providing audit ability and traceability of artifacts and their inter-relationships across the lifecycle across platforms
 - ▶ Stronger workflow capabilities: state changes, approvals, notifications, integration with other lifecycle phases including requirements & testing



Agenda

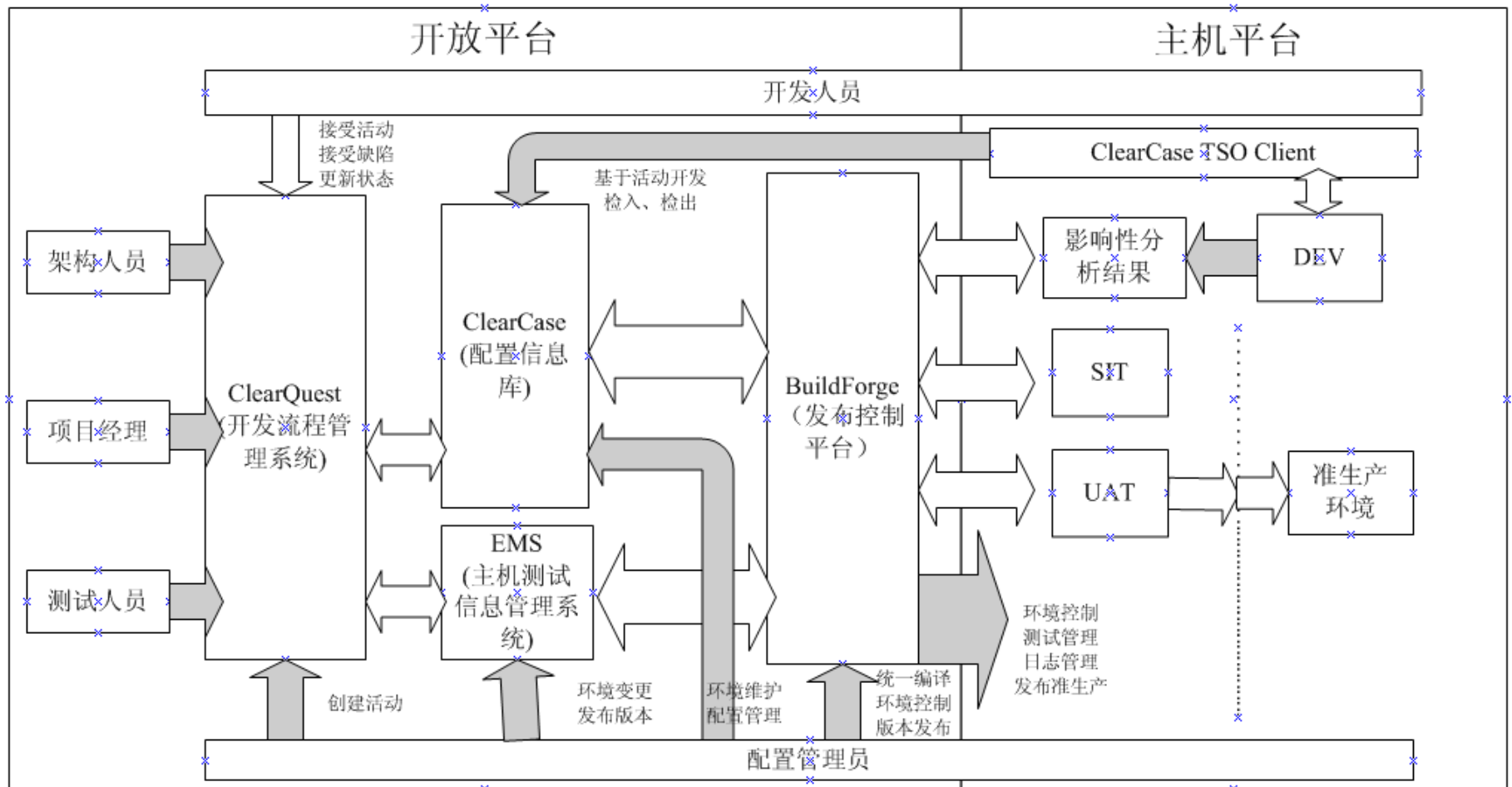
- z/OS Software Configuration Management Challenges

- IBM Solution
 - ▶ Solution Overview
 - ▶ Architecture and Deployment Model
 - ▶ Backend Technical Detail

- Case Study and Implementation Discussion



Solution Architecture



Involved Tools

- ClearCase – Configuration Management
 - ▶ Version Control
 - ▶ Parallel Development Support
 - ▶ Baseline Management for Composite Applications
 - ▶ Seamless Integration with Change Management and Build&Release Management

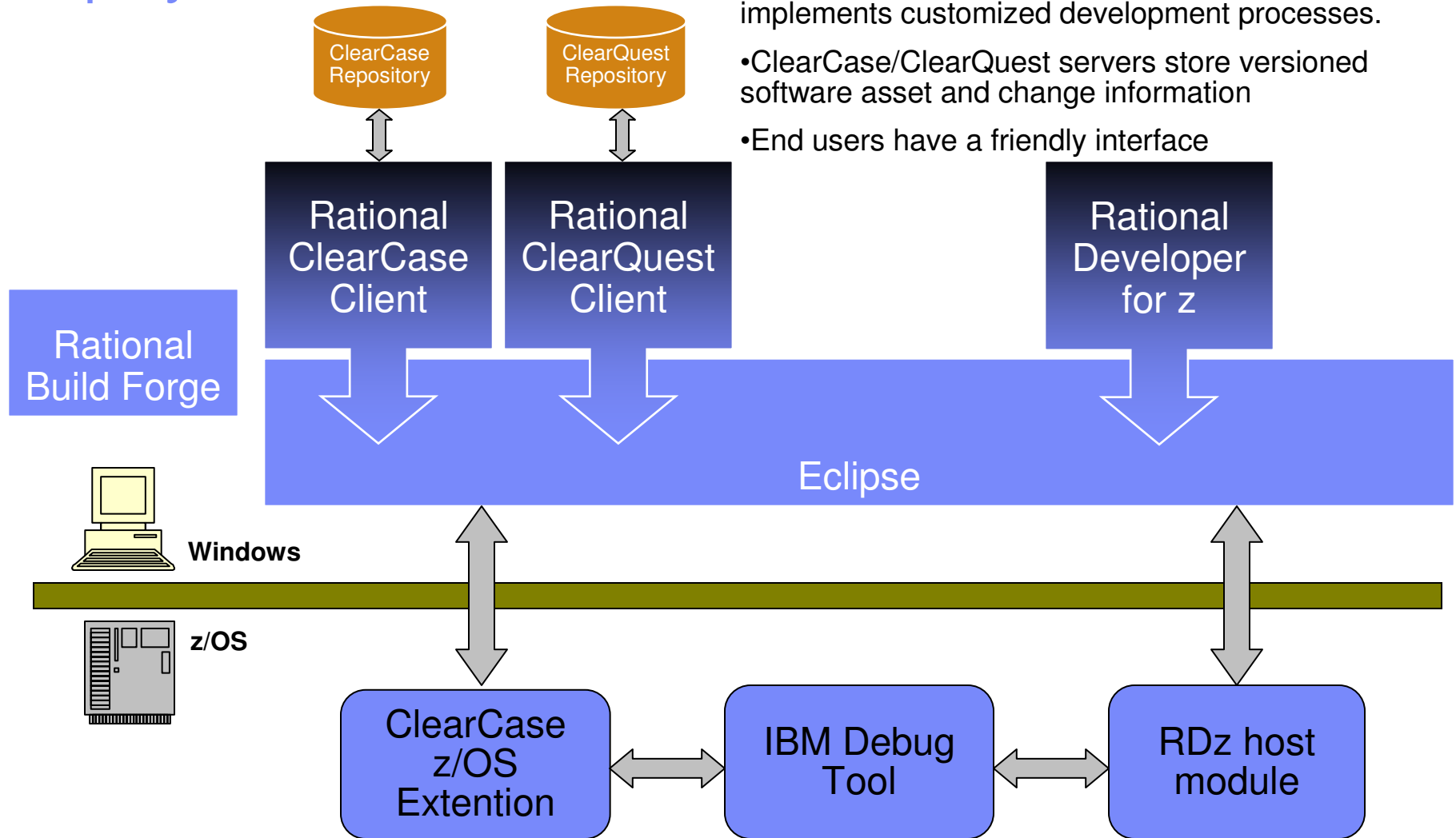
- ClearQuest – Change Management
 - ▶ Request/Development task/Change/Defect Management
 - ▶ Flexible and Customized Development Work Flow

- BuildForge – Build and Release Management
 - ▶ Integrate and Automatic Compile and Release Process
 - ▶ Unified Manage Cross-Platform Build Process

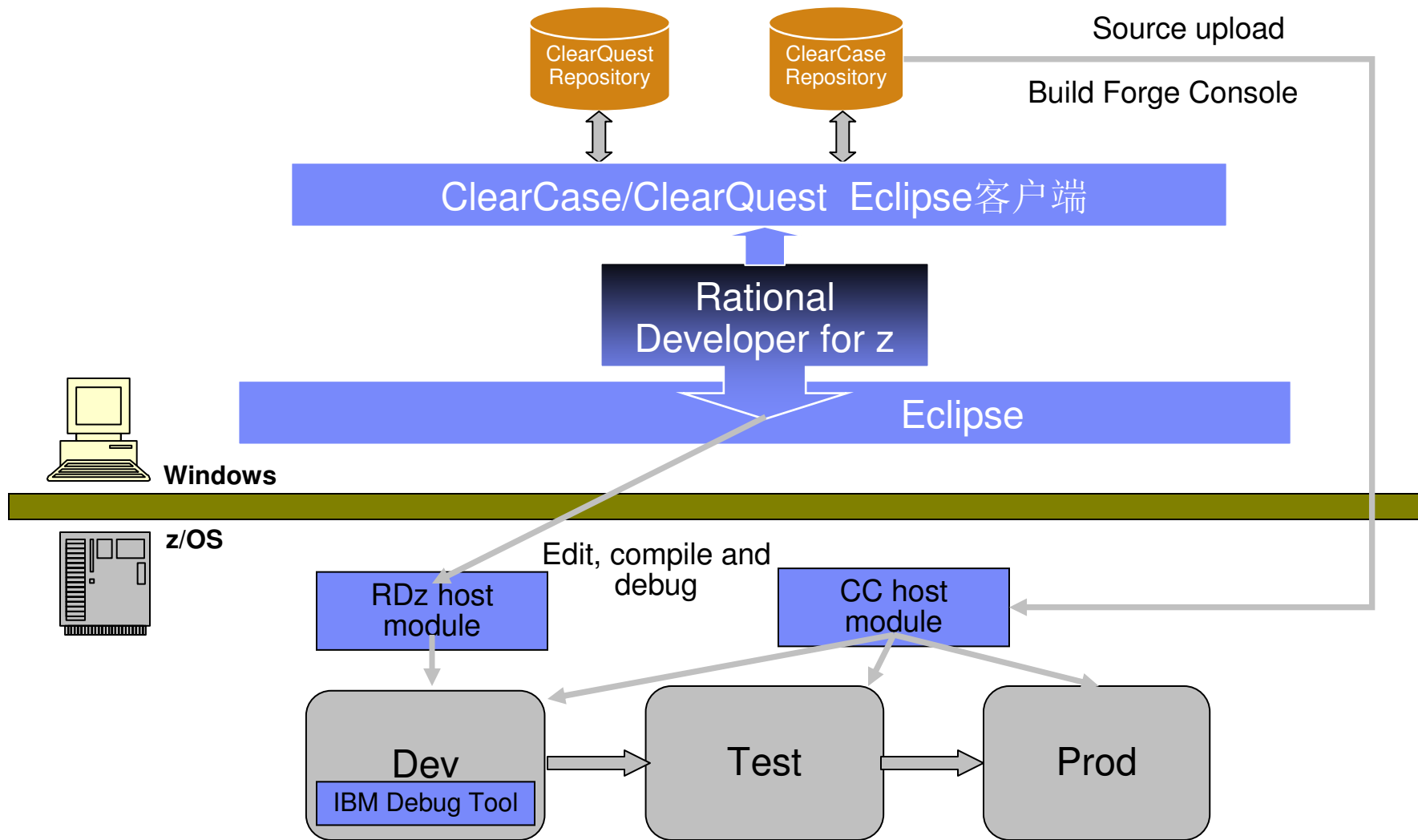


Deployment Model

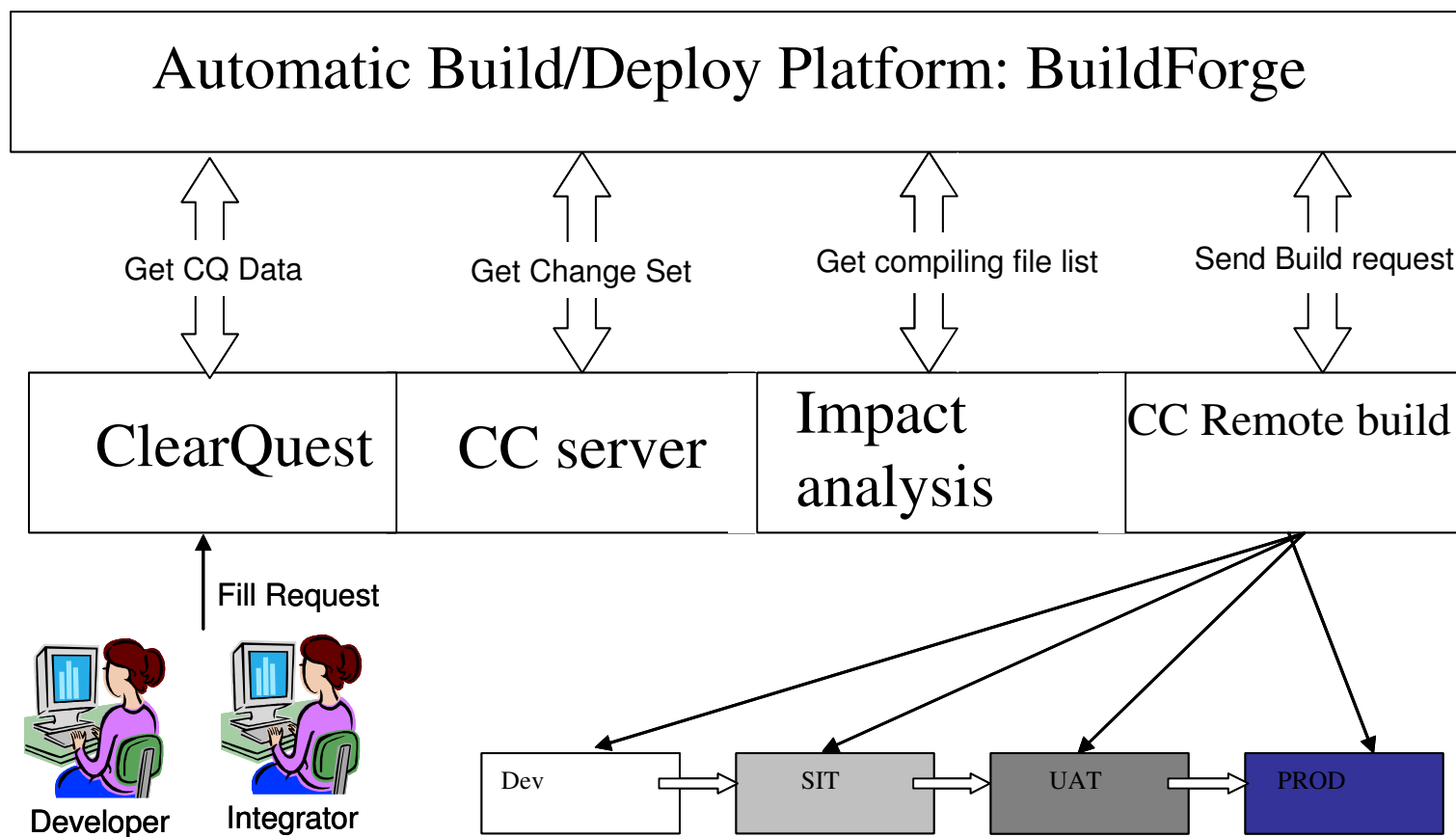
- Unified integrated development environment based on three Eclipse plug-ins, which automatic implements customized development processes.
- ClearCase/ClearQuest servers store versioned software asset and change information
- End users have a friendly interface



Interaction Between Software Assets and Development Environment

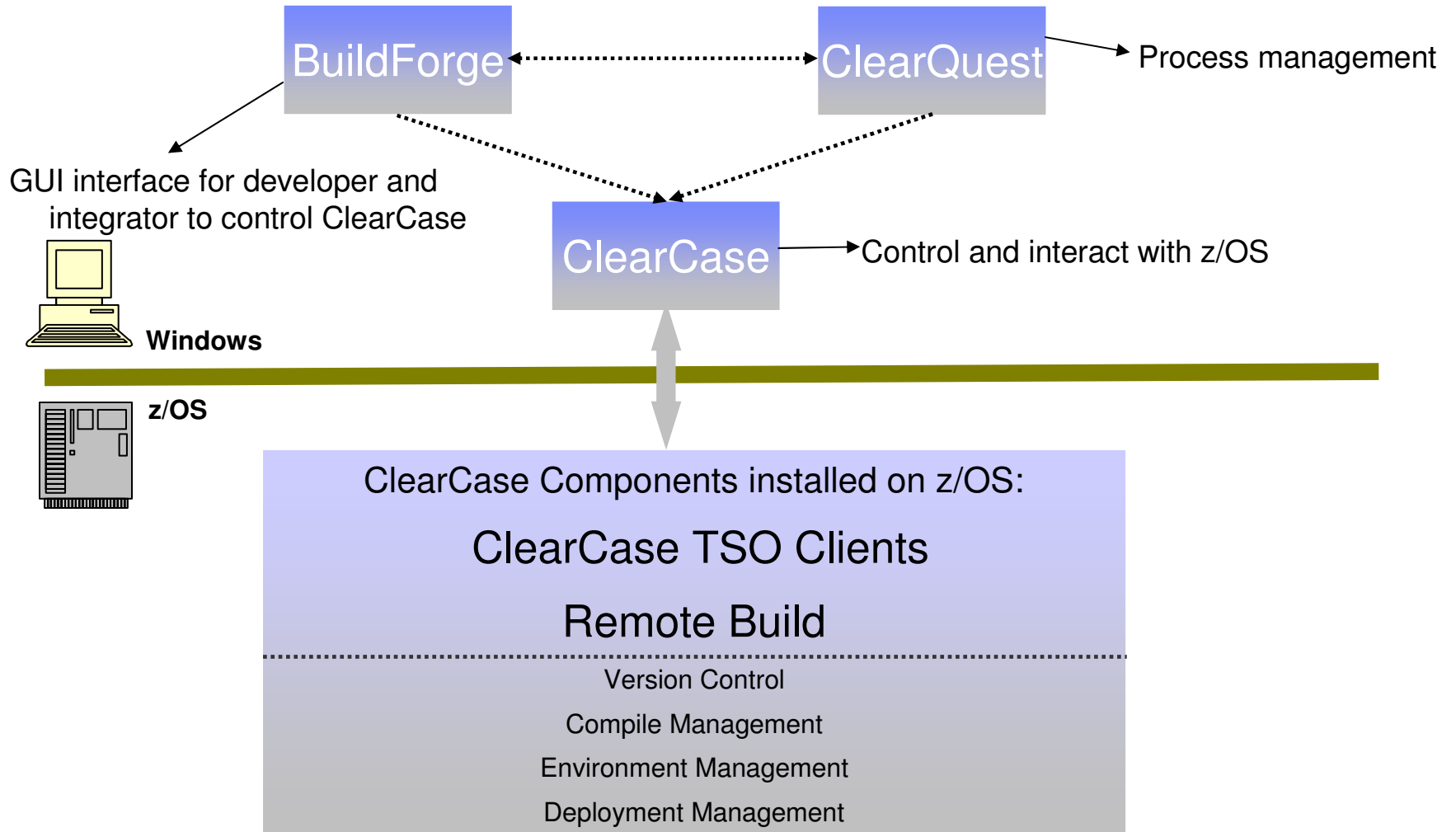


Automatic Build Architecture



BuildForge and ClearCase/ClearQuest implement monitored automatic build/release platform

Interaction between ClearCase and z/OS



ClearCase TSO Client

```
Command ==> _____  
  
          Rational Clearcase Mainframe Connectors TSO Client  
          Select Action for Member:      2.2.1  
  
Option ==> _____  
  
          1. Check Out           4. Compare With Previous Version  
          2. Check In           5. Show Version Tree  
          3. Undo Check Out     6. Add to Source Control  
  
          Comment for check out and check in option:  
          _____  
  
          PF01=ClearCase Help           PF03=Return to ViewFile List  
  
MA  b                                     09/014
```

- Access ClearCase repository from ISPF environment
- TSO users can use ClearCase in their comfortable environment



Two Usage Models

- On host development via ClearCase TSO client
 - ▶ Keep green screen development convention
 - ▶ Integrator works on distributed side to do build and deployment management

- Off host development via Rational Developer for System z(RDz)
 - ▶ Dramatically save CPU cost by migrating development activities to distributed side
 - ▶ Increase developers' performance by using Enterprise Modernization tool RDz
 - ▶ Take advantage of full ClearCase' power by seamless integration with RDz

- Usage model could be customized according to requirements



Agenda

- z/OS Software Configuration Management Challenges

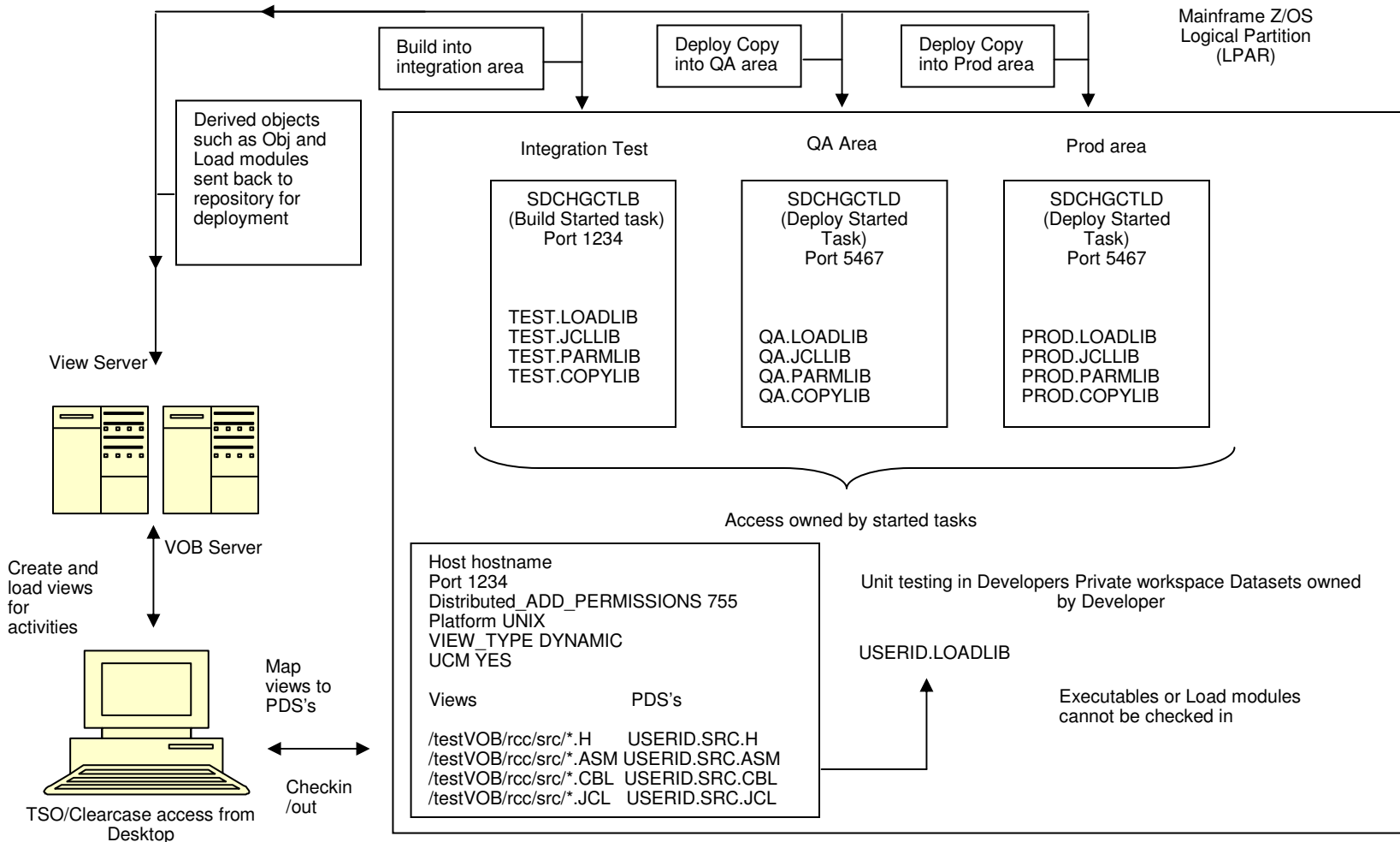
- IBM Solution
 - ▶ Solution Overview
 - ▶ Architecture and Deployment Model
 - ▶ Backend Technical Detail

- Case Study and Implementation Discussion



What is ClearCase Z/OS Extensions?

- Interface between ClearCase and Mainframe



Element Level Attributes

- Element level attributes – What components or the pieces and parts on how the executable is created
 - ▶ PROG1
 - DB2
 - IMS
 - CICS
 - MQ Series
 - LANG



ClearCase and Element level attributes

- Attributes are assigned at either migration or when adding to source control
- Not limited to 8 characters additional metadata can be stored
- Attributes Naming Convention
 - ▶ Online = Y or N
 - ▶ LANG = COB, PL1, ASM
 - ▶ DB2 = Y or N
 - ▶ CICS = Y or N
 - ▶ IMS = Y or N
 - ▶ MQ = Y or N
- No limit to what can be done
- Build Engine interrogates element level attributes to determine build components and the path to take through the Build Engine



Project Level Attributes

- Project level attributes – Defines where the project is built or deployed to the Z/OS infrastructure used in combination with CC environment variables.

- Projects level attributes contain
 - ▶ Syslib concatenation
 - ▶ Copybook libraries
 - ▶ target load libraries
 - ▶ DB2 subsystem name
 - ▶ CICS region
 - ▶ IMS region



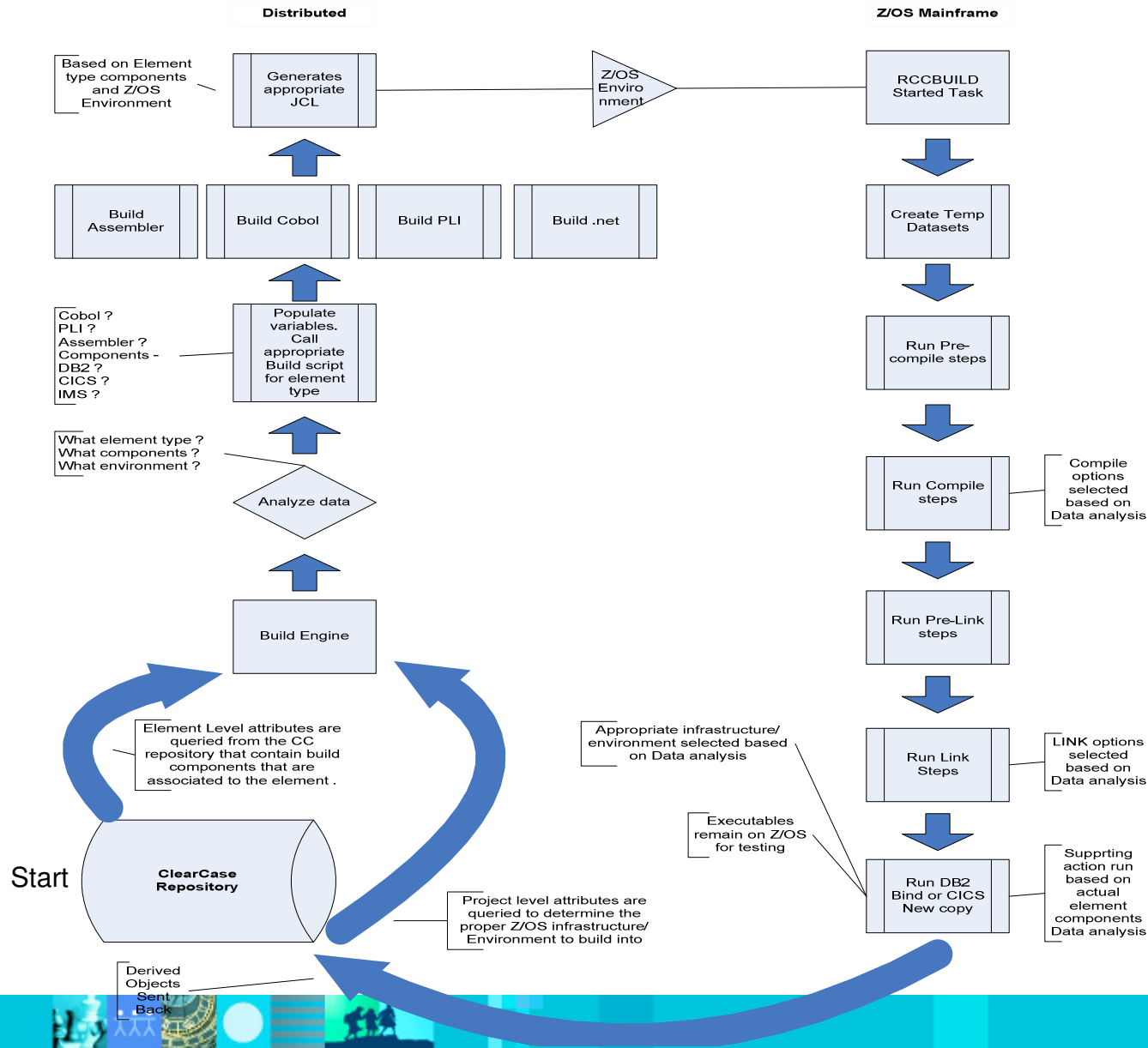
CC Environment Variables

- Environment Variables – used to map ClearCase architecture to the Z/OS infrastructure base on naming convention used.

- Dev → UT → QA → PROD
 - ▶ Stream name
 - ▶ User name
 - ▶ View name
 - ▶ Delivery to Stream name
 - ▶ Delivery from stream name
 - ▶ Element version



Remote Build Technology Detail

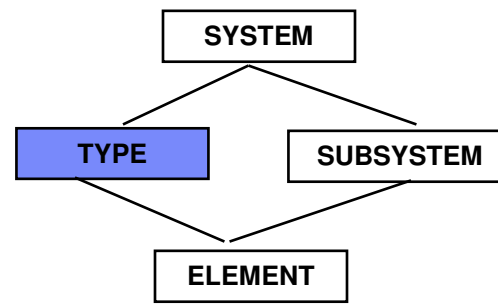


ClearCase CCz Build Engine

- Processor/CCz
 - ▶ CCz version of JCL (e.g. BCL Build control language) with keywords and utilities invoked by ClearCase actions
- Processor Group/CCz
 - ▶ Element level attributes set of build/deploy components required to manipulate elements of a particular type (e.g. Cobol, JCL, PL1)
- PERL – condition logic used to make choices about how to build and deploy
 - ▶ Endeavor processor language, not as robust as PERL
- BCL – PERL driver creates the correct BCL based on Build or Deploy components
 - ▶ Identical to Endeavor to execute mainframe utilities
- RCCBUILD command sends the work to the mainframe
 - ▶ Runs the work, executes the utilities, similar to Endeavor's SCL software Control language



Processors/CCz Build Engine



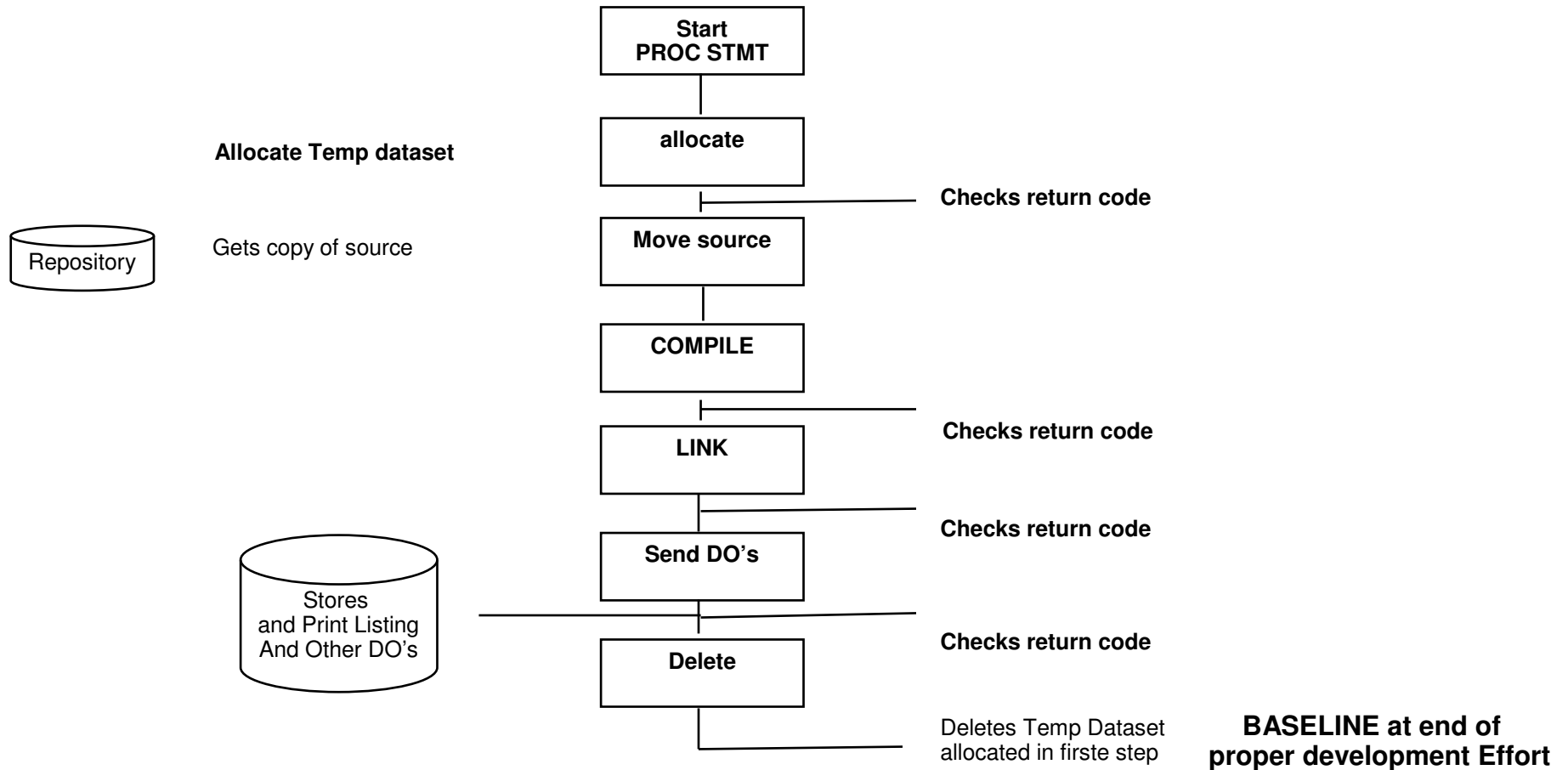
Can be organized in the same way by system, subsystem, Type relationship by storing metadata (attributes) search chain within structure of project or subsystem.

Build Engine Managed build from source element (BUILD,COMPILE)

- MOVE** Promote (copy) outputs through the life cycle
only effects at promotion time based on CQ action or BL promotion level
- DELETE** Deletes outputs created by build Engine only in MF environment
- COPY** Movement of source (non-compiled) simply copies source to appropriate MF environment



How does a CCz Build Engine work/Structure



Agenda

- z/OS Software Configuration Management Challenges

- IBM Solution
 - ▶ Solution Overview
 - ▶ Architecture and Deployment Model
 - ▶ Backend Technical Detail

- Case Study and Implementation Discussion



Case Study and Implement Discussion

- PRC Case Study
- Implement Discussion



