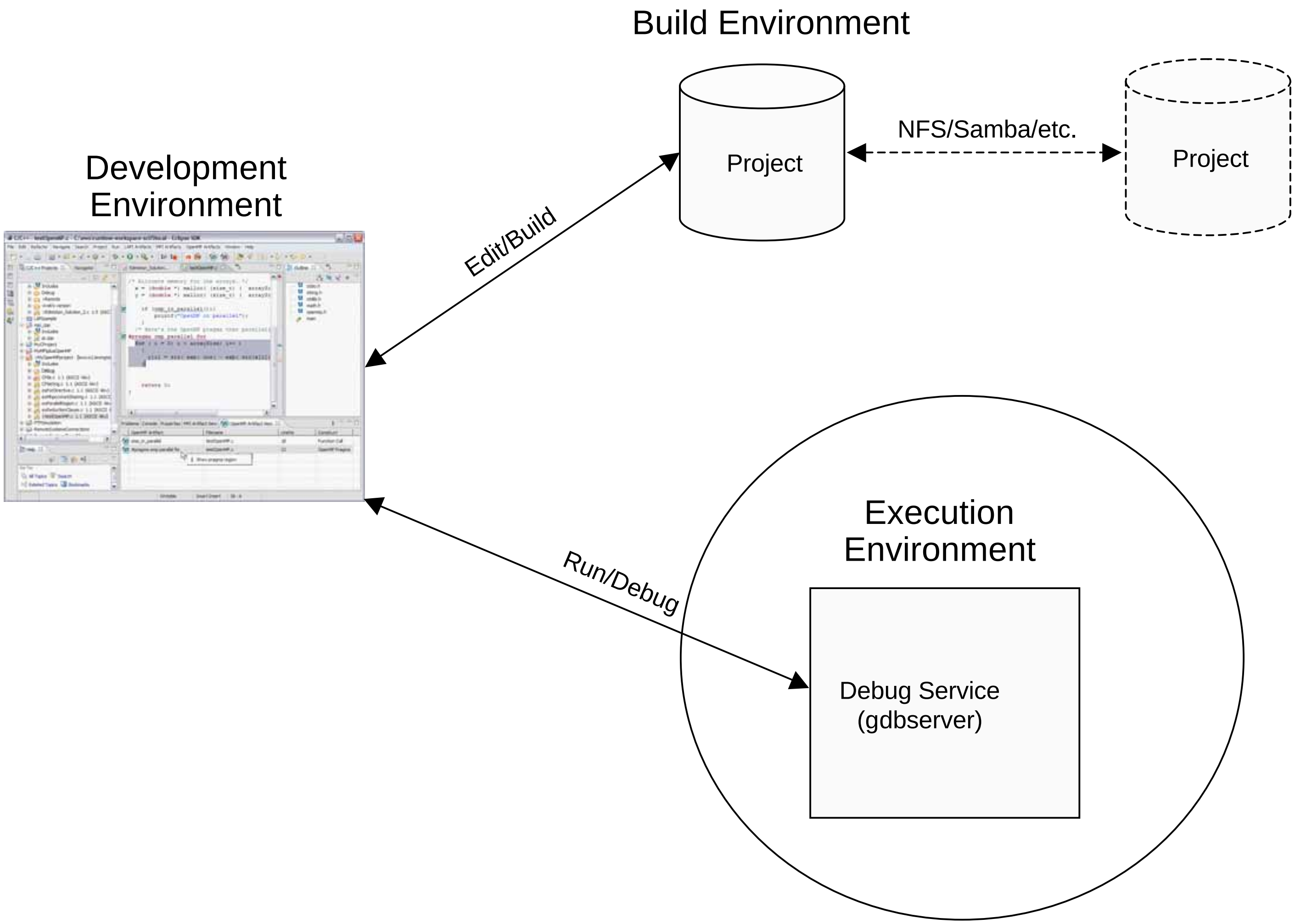
Remote Development with CDT

Greg Watson
Chris Recoskie
IBM

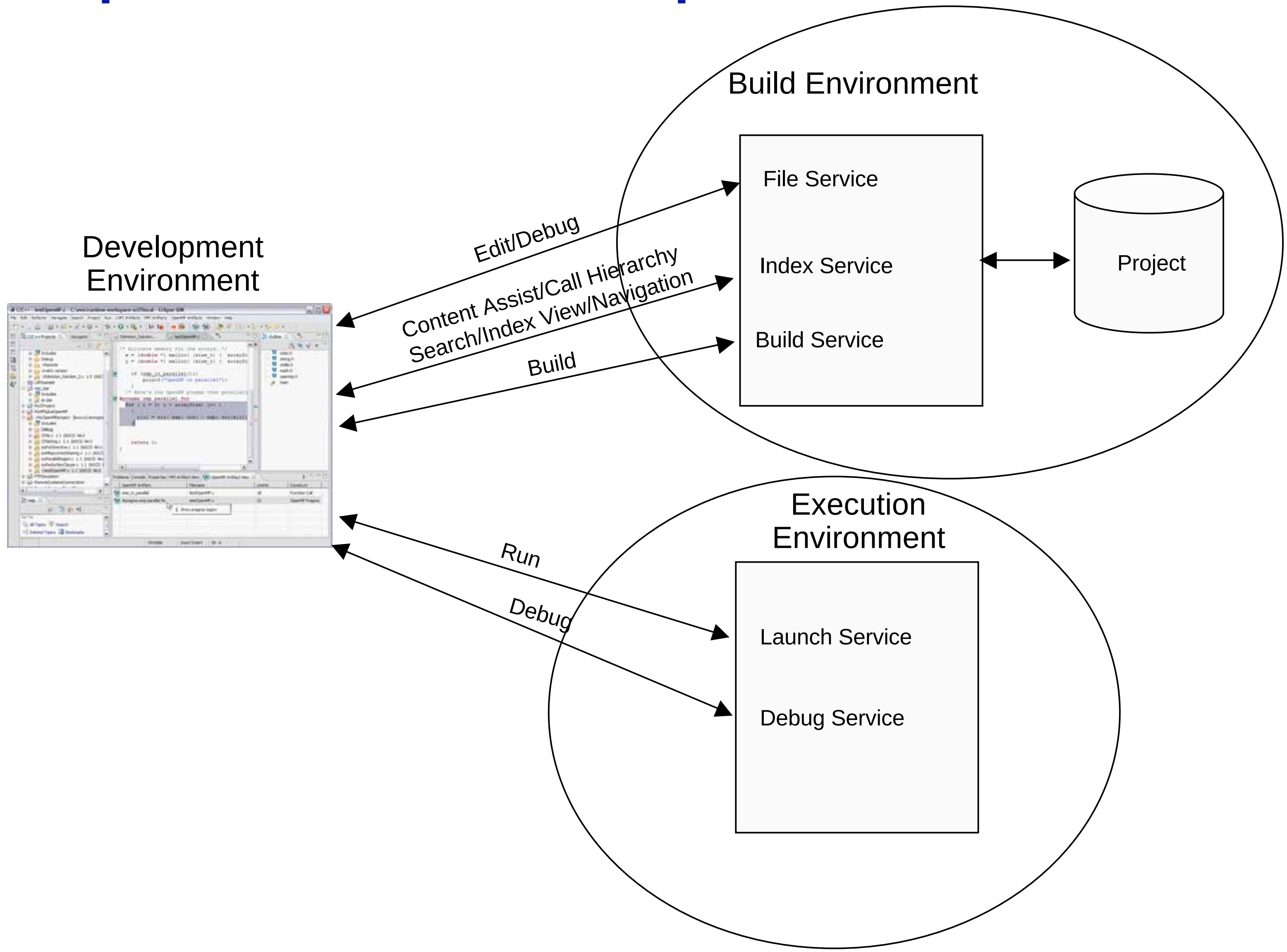
Existing Remote Development Environment



Existing Remote Development Status

- Editing (including content assist, call hierarchy, search, index view, navigation, etc.) assumes local filesystem
- Build (managed and standard) assumes local filesystem
- Cross-compiler must be used to support different architecture
- Can separate build environment using NFS, Samba, etc., but all operations still performed locally
- Remote launch/debug using gdbserver only

Proposed Remote Development Environment



Requirements for RDE

- Goal: provide a seamless and transparent remote development environment
- Abstraction layers so that key services can be provided locally or remotely
  - EFS for file services
  - Remote search/index service to support efficient content assist, call hierarchy, search, index view, navigation, etc.
  - Remote build service to support environments where cross-compiling is not desired/practical
  - Remote launch service to support running on a variety of architectures
  - Remote debug service to support non-gdbserver debug environments
- Local service implementations to preserve status quo
- Reference implementations against remote service provider (e.g. RSE)

Requirements for CDT

- Minimal impact on existing performance
- Preserve existing workflows and UI interfaces
- No added dependencies
- Out-of-box experience should not be degraded

EFS Integration

- IPath no longer refers to an actual filesystem object; URI does this now
- IResouce.getLocation() can return null
- Use URI and IFileStore instead of IPath.toFile()
- Need to use IResource whenever possible
  - Can no longer use File objects or java.io.File unless it is known that the resource is local
- A Path object can be constructed with any String, however a URI can only be constructed with a valid URI
- How to deal with
  - Case sensitive file system
  - Interpretation of characters in file names (e.g. ':' under Windows vs. Unix)
  - Mapping remote paths into workspace locations (e.g. ErrorParser)
- Responsiveness
  - Must assume any file operation could be long running or can fail

EFS Integration (cont…)

- 3,173 uses of IPath
- 793 uses of Path
  - 20 uses of Path followed by isValidPath()
- 595 uses of java.io.File
- 140 uses of IPath.toFile()

EFS Conversion Progress (~2 weeks)

- Added a filesystem combo to new project wizard when 'use default location' is unselected
- Removed File dependencies from
  - {GNU,Posix}Makefile classes
  - Makefile creation
  - Checksums
- IPath to URI
  - ProblemMarkerInfo
  - Breakpoint problem markers
  - Scanner config marker generator
  - Build output console parser
  - Build directory lookup
  - CdtMacroSupplier
  - PDOMIndexerTask
- Started conversion of ErrorParser
  - Issues still need to be addressed (e.g. Cygwin)

Demo…

Remote Indexing

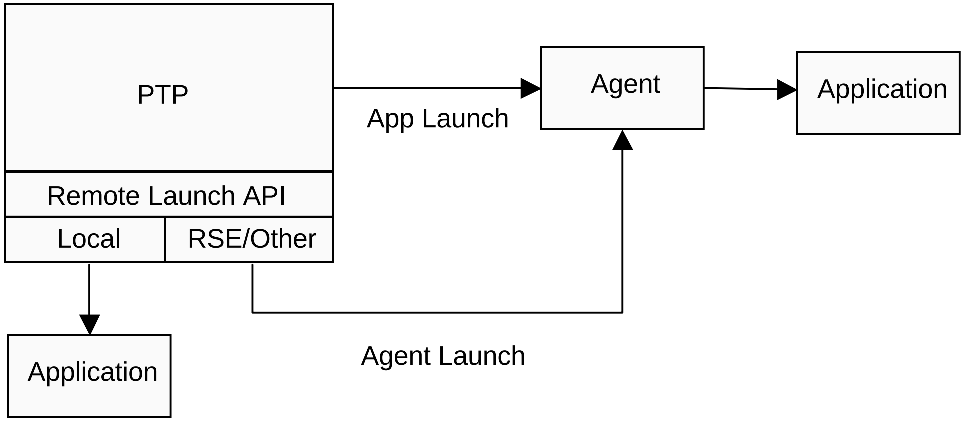
- Remote indexing framework will be moved to PTP or new RDT
- Some changes still needed in CDT to allow code-reuse of views
  - Model-View-Controller pattern
- New remote versions of views to be created in PTP or RDT
- Remote back-ends for index based services to be contributed to PTP or RDT, providing a reference implementation of remote services using RSE.

Remote Build

- Phase 1:
  - Remote version of standard make builder will be created that uses RSE shell protocol to invoke the build
  - The usual error parsers, scanner discovery will be run on the output
  - Planning to do this for Ganymede
- Phase 2:
  - Remote service for build so that Scanner Discovery can be run out output to feed remote indexer.
- Phase 3:
  - Remote Managed Build with new Remote Builder (similar to internal builder)

Remote Launch

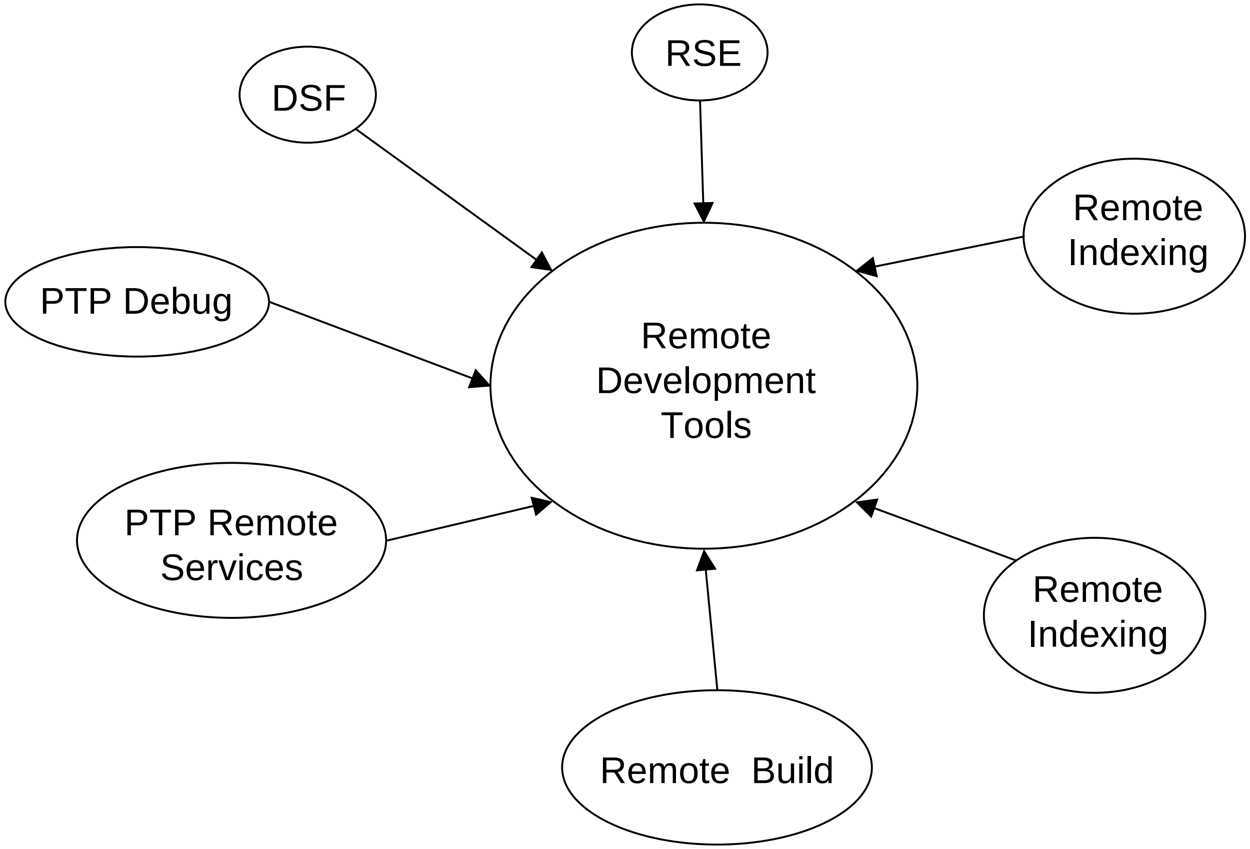
- PTP provides remote services abstraction
  - Allows different remote service providers to be used
  - Currently RSE, but a lightweight provider is being contributed
- Using this to launch PTP agent
- Assumption is launch/debug are remote



Remote Debug

- DSF
- PTP
- CDT/gdbserver

Remote Development Tools Project?



RDT Progress

- IBM and HP have been holding discussions about proposing a Remote Development Tools project (RDT)
- Charter would be to handle remote development tools frameworks and solutions for various languages
  - Plan to collaborate on remote development tools solution for C/C++ using CDT and RSE
  - Remote tools for Fortran?
  - Remote tools for Java?
- Currently trying to gather momentum leading up to a project proposal at Eclipse.org