Synchronized Projects Internals and Operation

John Eblen jeblen@acm.org

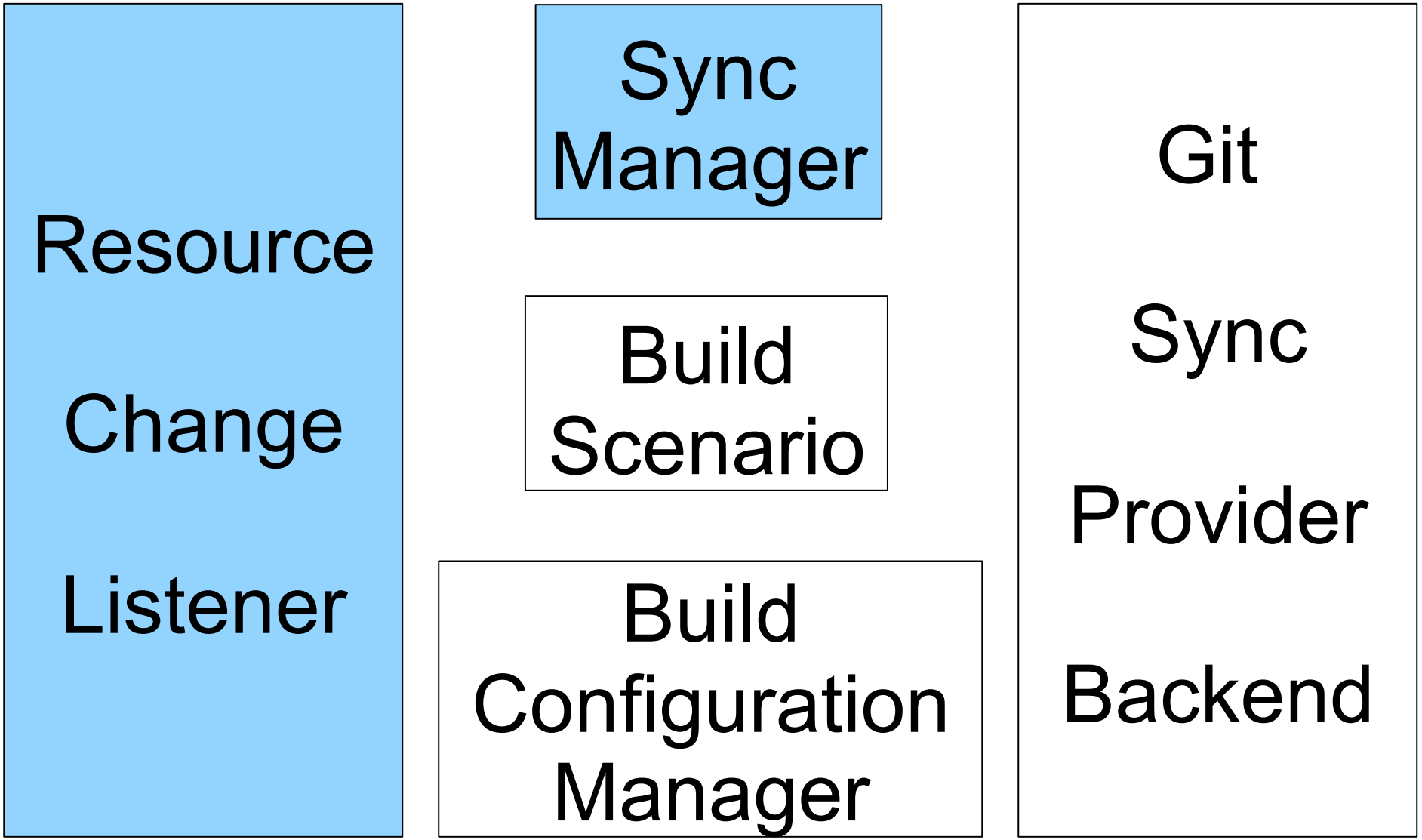
September 19, 2012

PTP User-Developer Workshop

Outline

- Five core “classes”
  - Resource change listener
  - Sync manager
  - Build scenario
  - Build configuration manager
  - Git backend
- Adding a new sync service provider
- Other components
  - Build system
  - File filtering
  - Merge resolution

Core Components



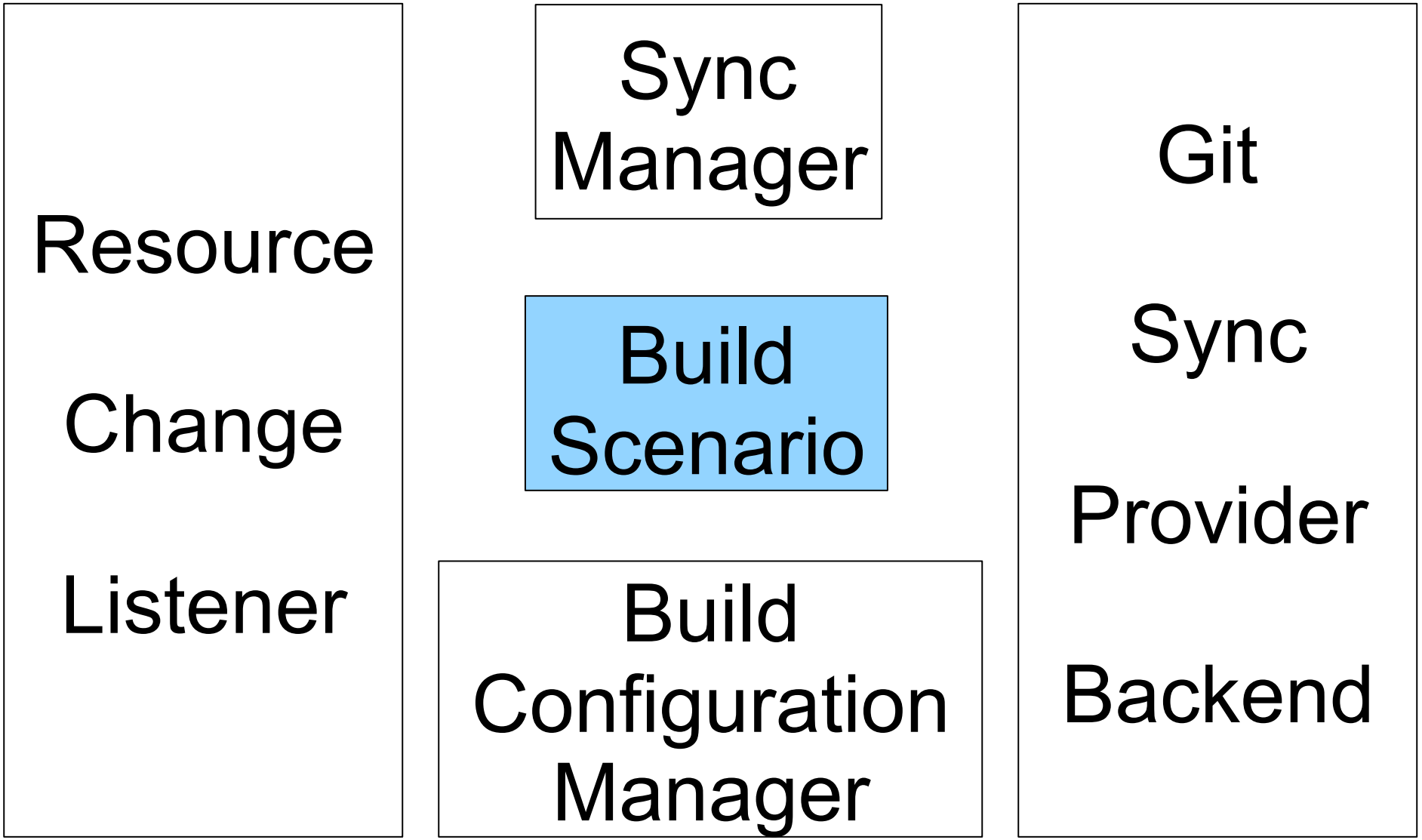
Resource Change Listener

- Interface between Eclipse and sync operations
  - Resource (file system) changes
  - Other events (i.e. post-build events)
- Menus provide another user-level interface

Sync Manager

- Static class – handles all sync requests
- Methods for accessing sync-related data
  - Sync mode
  - Auto-sync setting
  - Sync error handling
- Different types of syncs
  - Blocking or non-blocking
  - Active or all

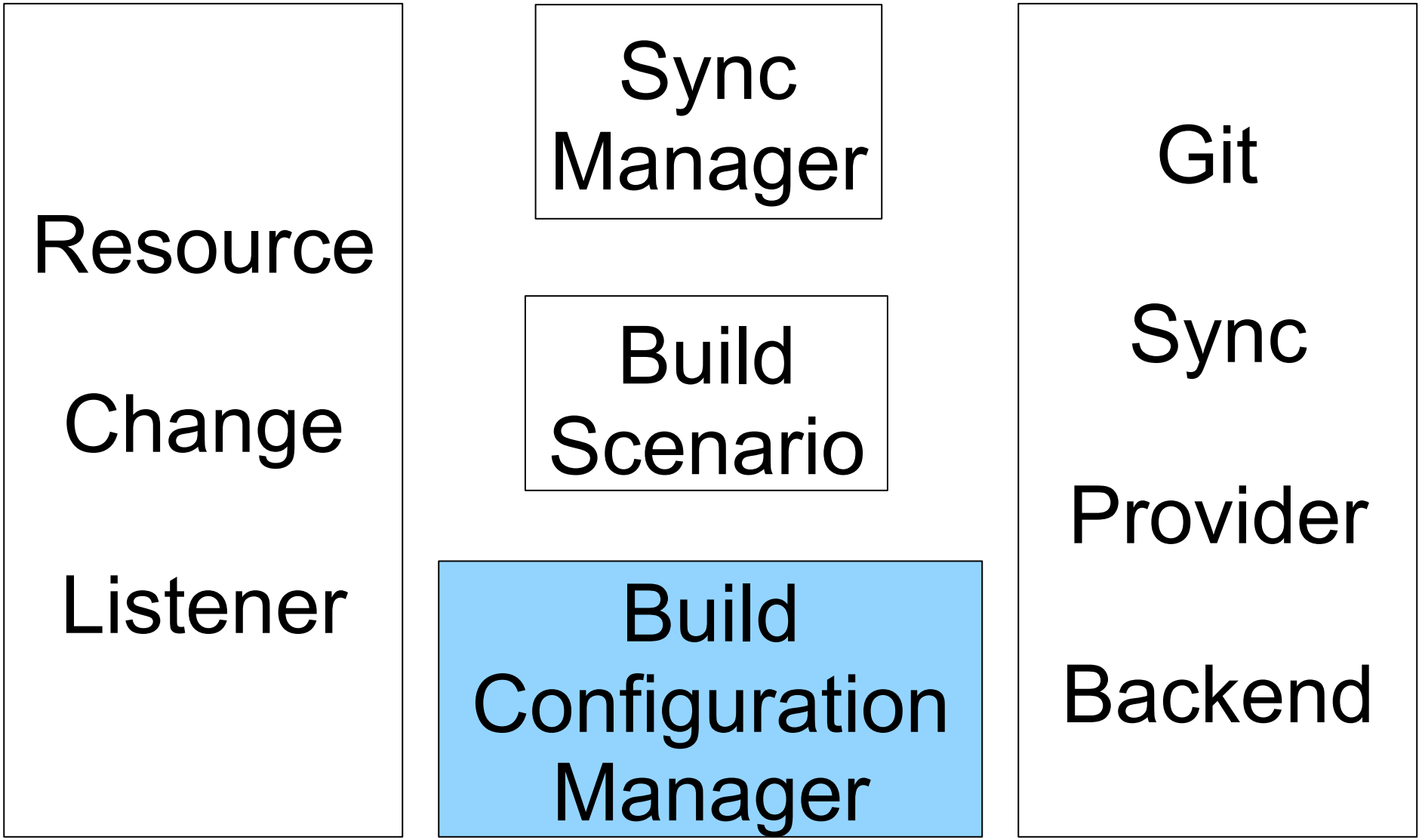
Core Components



CDT Build Configurations and Build Scenarios

- All synchronized projects are also CDT projects
- CDT allows each project to have multiple “build configurations”
- Synchronized projects add additional data
  - Connection
  - Location
- Allows building on multiple remote systems
- Data stored in build scenarios – used in several places

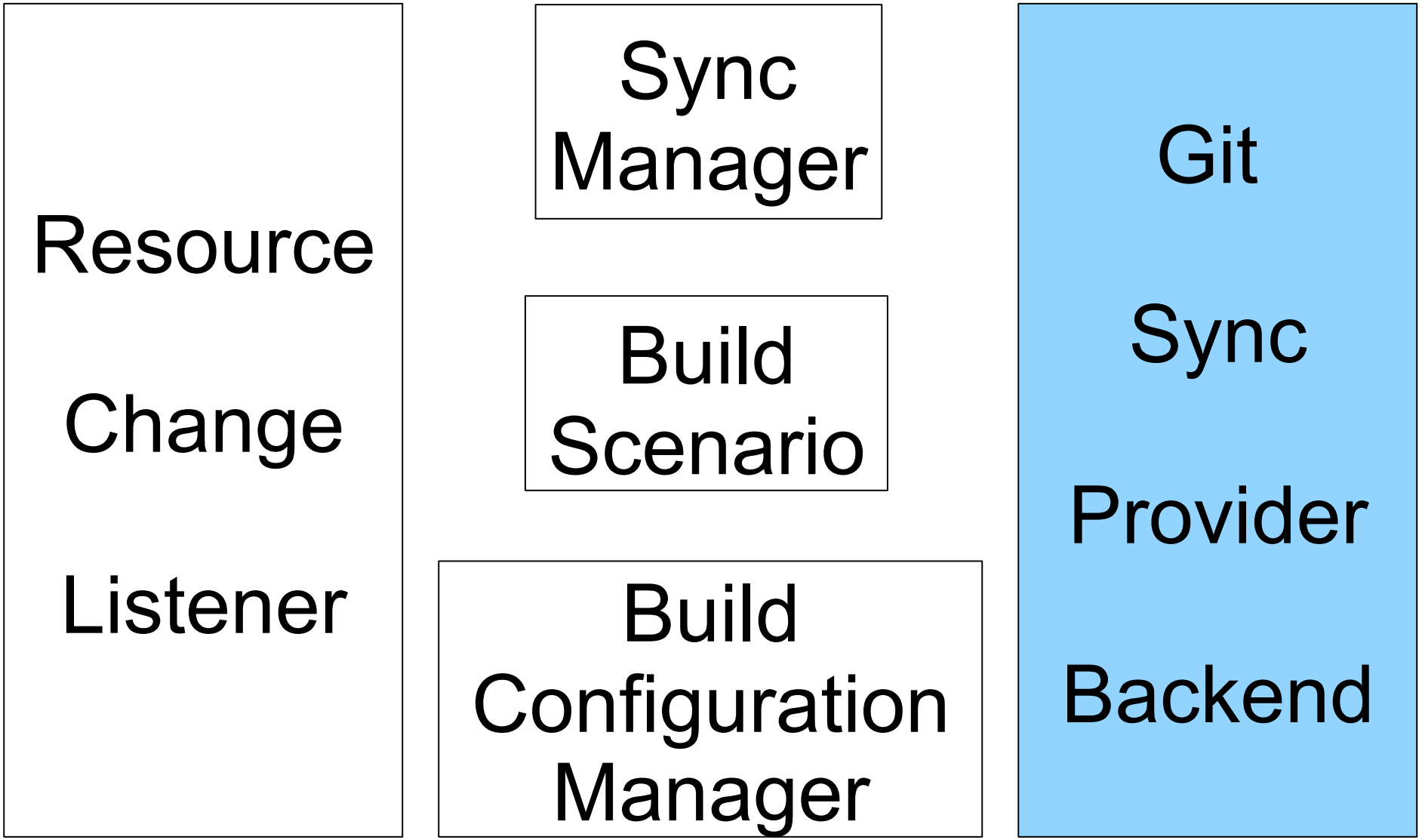
Core Components



Build Configuration Manager

- Singleton class – interfaces with CDT to handle build configurations
- Roles
  - Retrieve and save build scenarios
  - Controls access to sync service
  - Logic to create build configurations (used during project creation)
- Heavily refactored
  - Store data in CDT
  - Remove service configuration wrapping
  - Use a single sync provider

Core Components



Git Sync Provider Backend

- GitServiceProvider Role
  - Plug-in class that interfaces to Git backend
  - Does some bookkeeping
  - Do not sync filtered files (Bug 386528)
  - Optimizes sync calls
  - Should have no Git-specific code – simply forwards calls in several places
- GitRemoteSyncConnection
  - Uses JGit to do actual sync'ing
  - Provides repository information

Adding a New Sync Service Provider

- Add plugin-in extension and new class extending ServiceProvider and implementing ISyncServiceProvider
- BuildConfigurationManager should load all providers
- Add GUI elements so users can select provider
- Use syncProvider field of BuildScenario

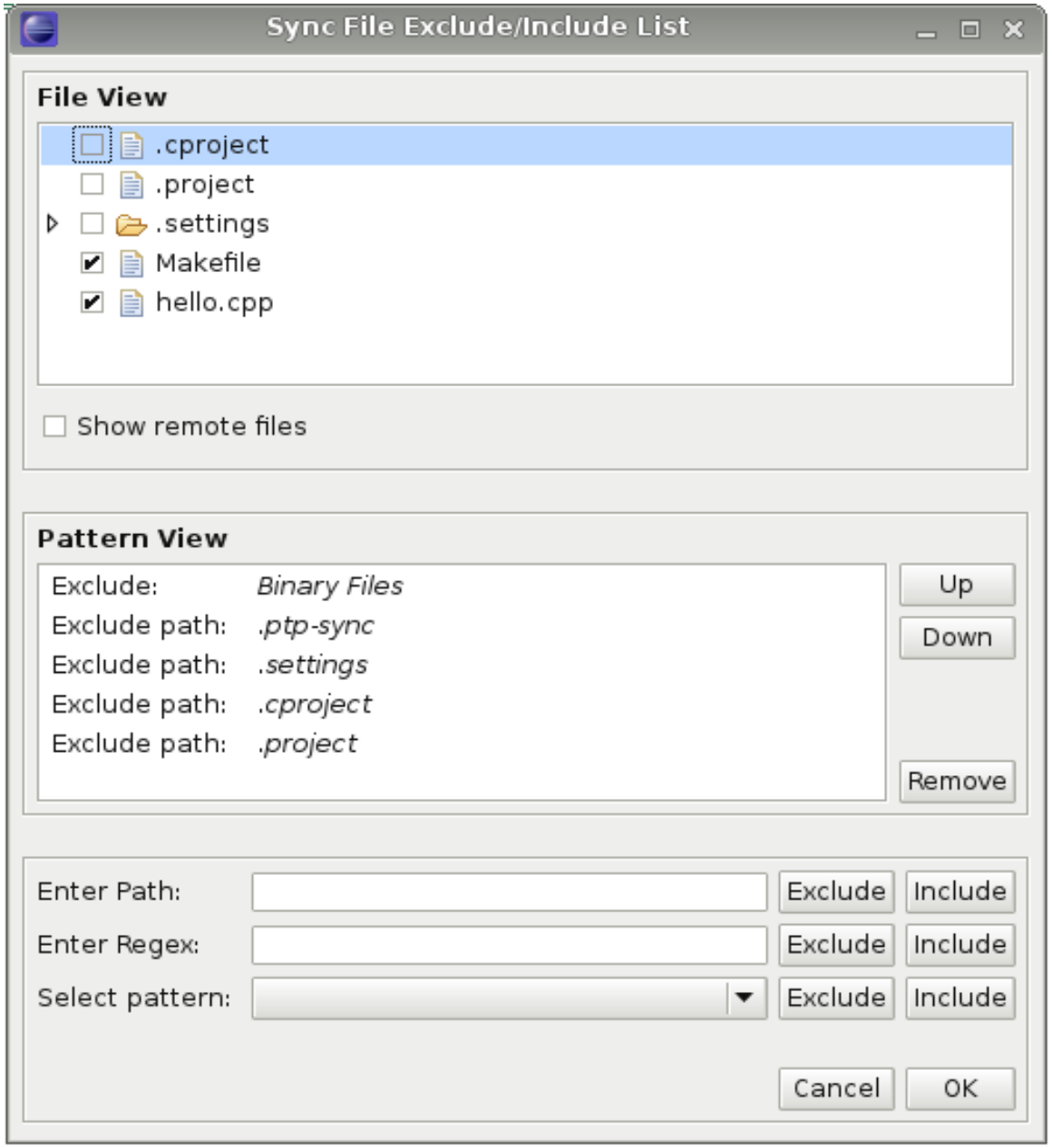
Build System

- Local builds are fine
- Remote builds require rerouting build command to remote
- CDT extension point – declare builder with command launcher class
- SyncCommandLauncher uses build scenarios to reroute build command to appropriate remote location

File Filtering

- Each project has an associated file filter (a class) that is accessible through the SyncManager
- SyncFileFilter interface
  - getPatterns()
  - shouldIgnore()
  - functions to add, remove, and demote patterns
- Each “pattern” is a “ResourceMatcher” instance (base class that encapsulates pattern-matching logic)
- Three types (subclasses) currently
  - Path matcher
  - Regex matcher
  - Binary matcher (removed for SR1)

File Filtering UI



Merge Resolution

- Sync service provider reports merge-conflict information
  - Conflicting files
  - Conflicting file parts
- Sync service provider also has functions for resolving merge conflicts
- Guard in GitServiceProvider prevents sync during merge conflicts