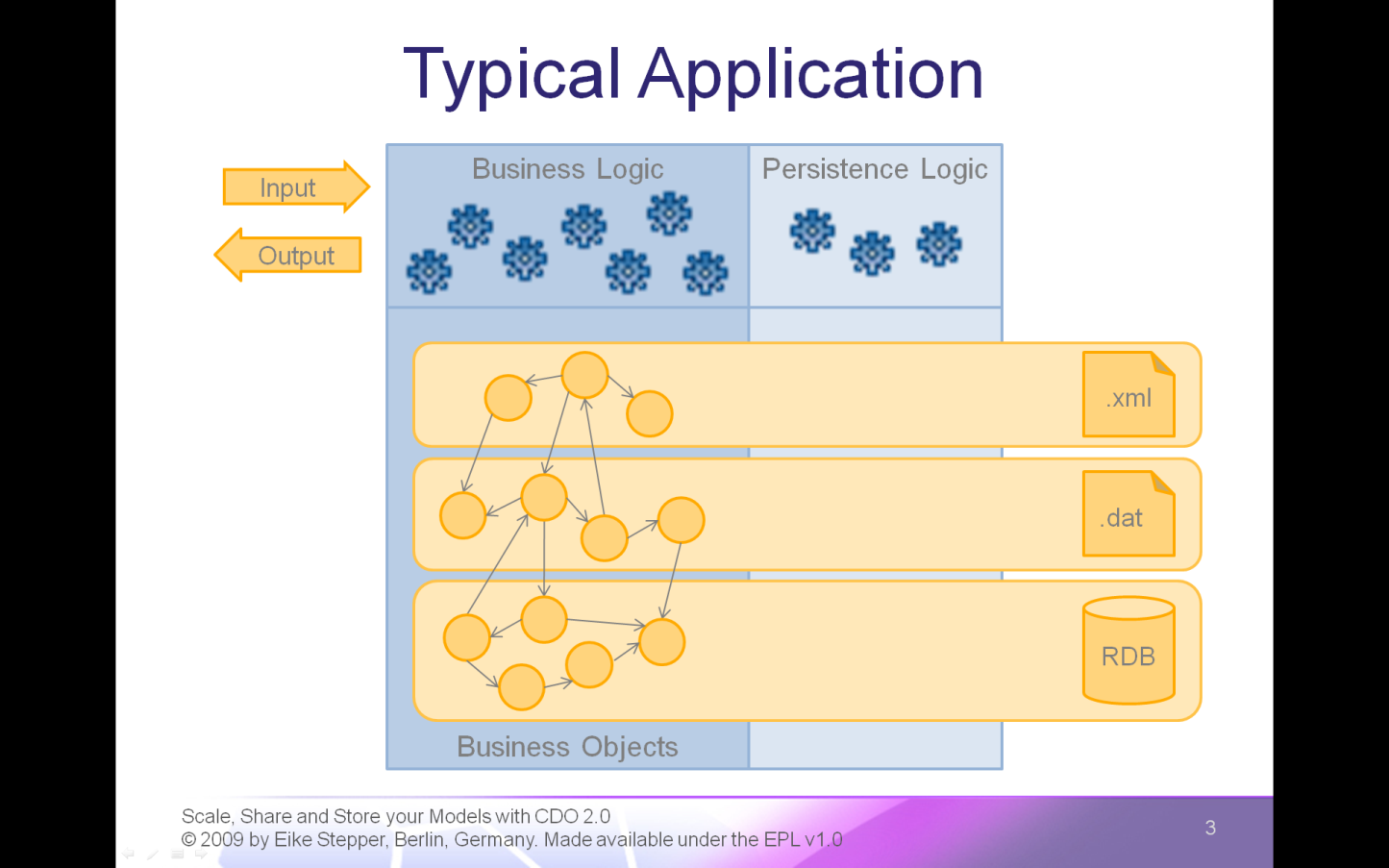
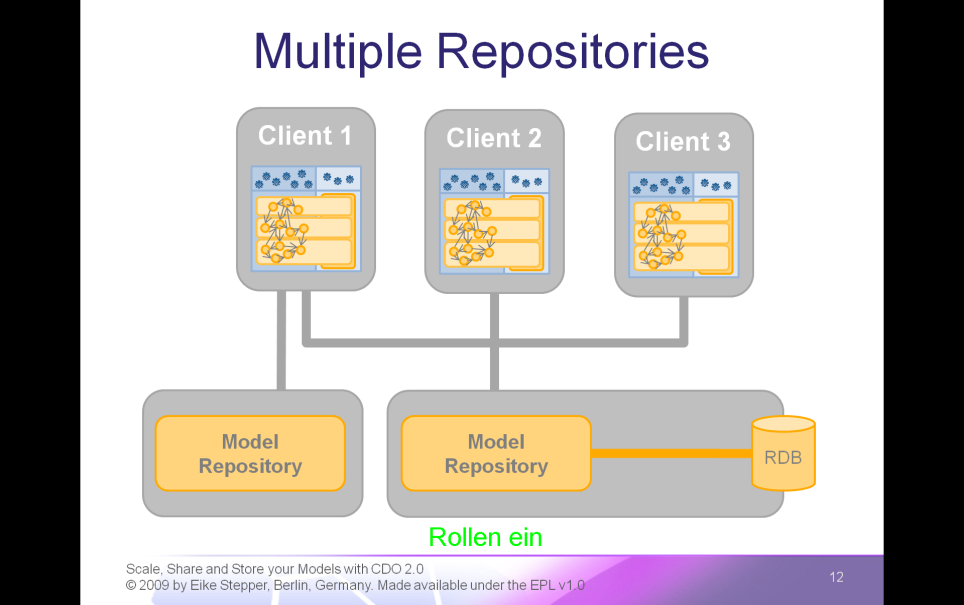
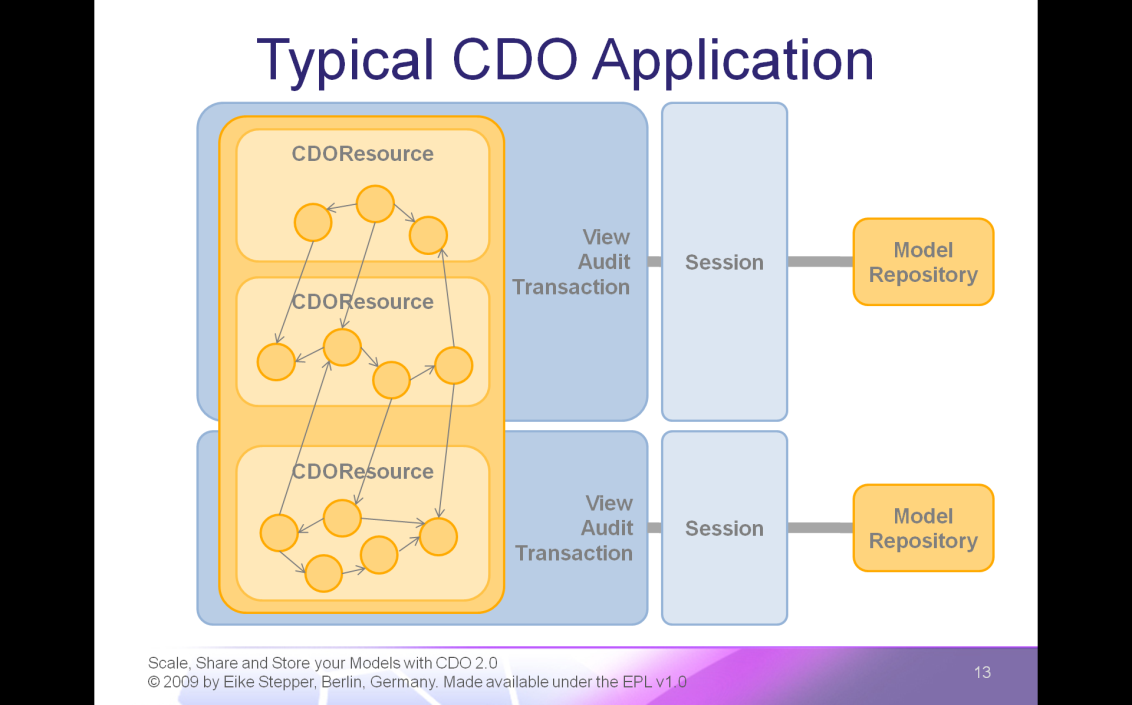
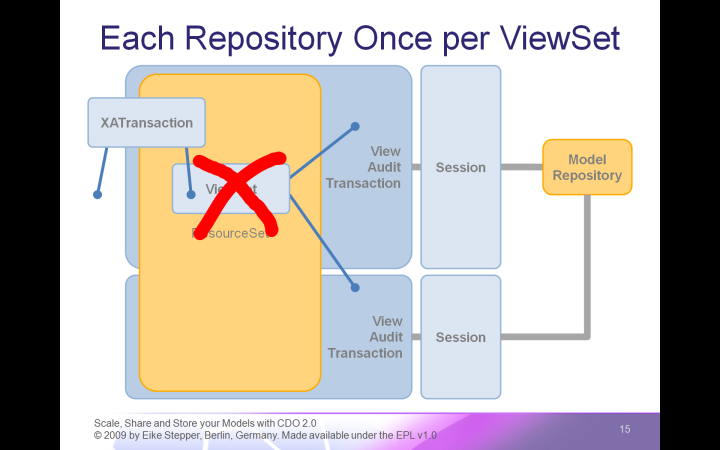
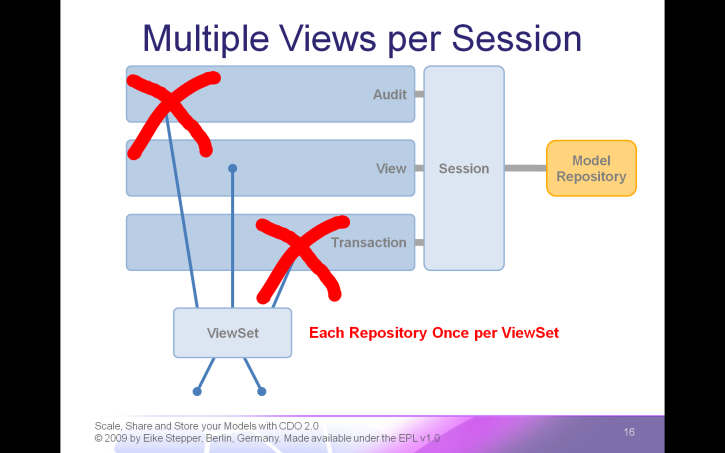
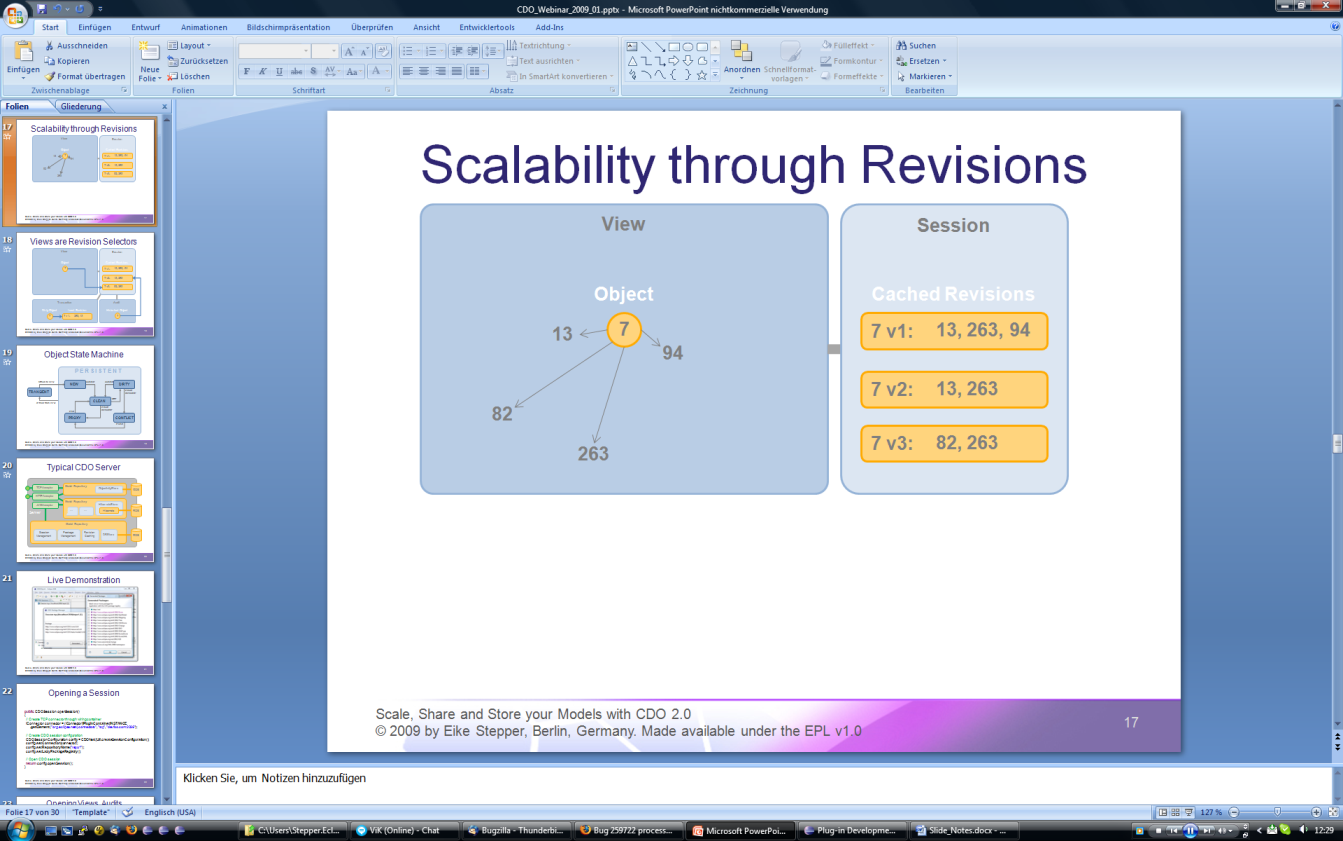
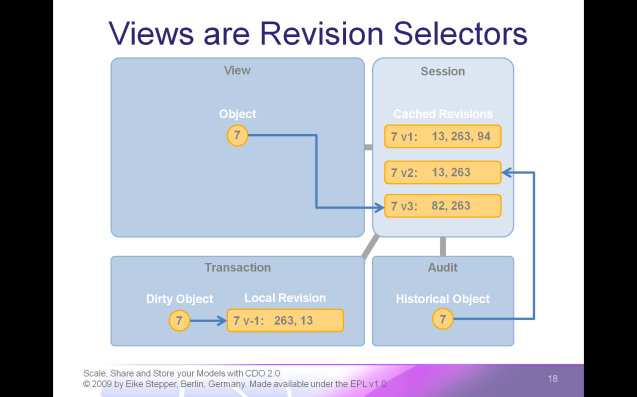
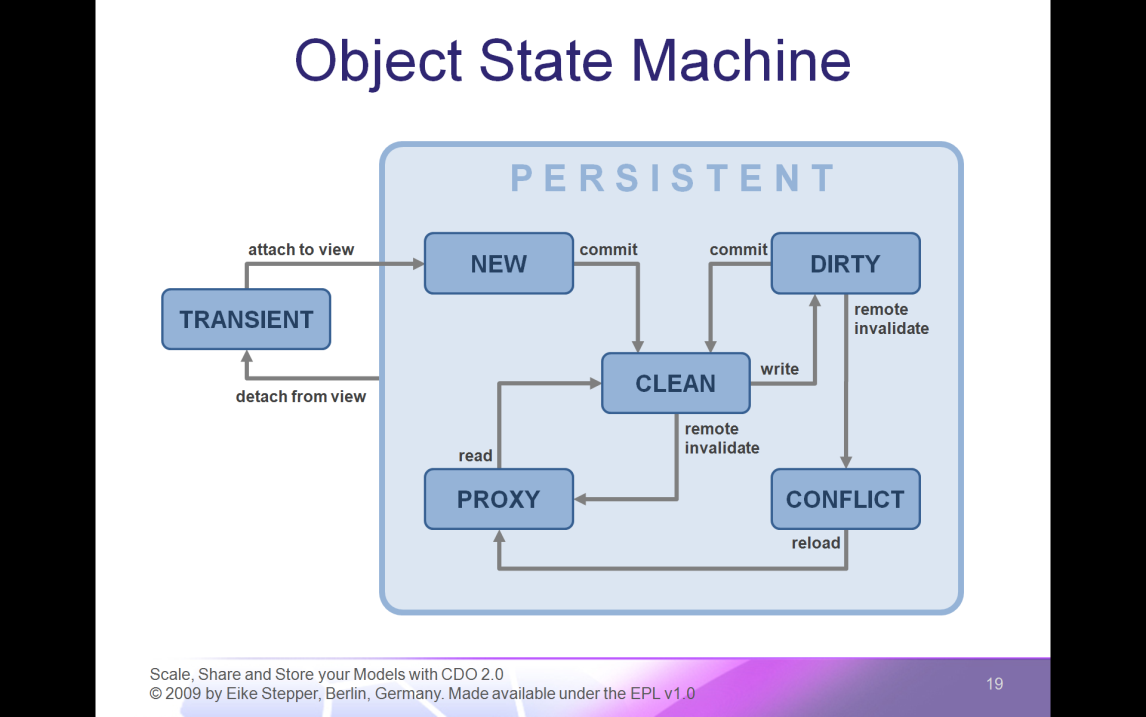
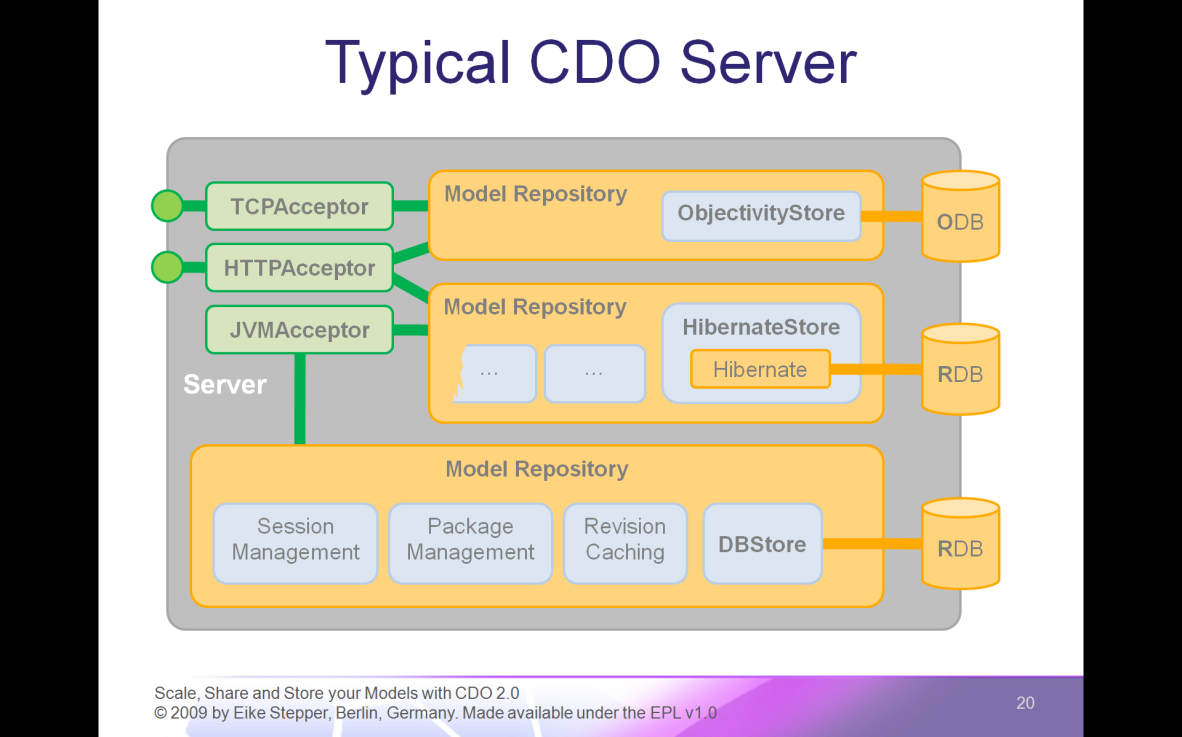
1. **Scale, Share and Store your Models with CDO 2.0**
2. **Agenda**
3. **Typical Application**
   * Input
   * Business Logic
   * Output
   * Business Objects
   * Persistent Backends
   * Orthogonal Persistence Layer
4. **Cross References by EMF**
5. **Model View Controller by EMF**
   * Input / Output (disappears)
6. **Business Logic is Yours**
   * Business Logic (disappears)
7. **Models Generated by EMF**
   * Business Objects (disappear)
8. **Persistence Handled by EMF**
   * Persistence Logic (disappears)
9. **Default Serialization by EMF**
   * XMLResource / BinaryResource (disappear), Question mark (appears)
10. **EMF Snippet**
11. **Typical CDO Deployment**
    * Network and Clients
    * Client Applications
    * Model Repository and Backend Connection
12. **Multiple Repositories**
    * Additional Model Repository
13. **Typical CDO Application**
    * Session
    * Session
    * View
    * View
    * CDOResource
    * CDOObject
    * CDOResource and CDOObject
    * CDOObjects
    * CDOResource and CDOObjects
    * Remaining CDOObjects
14. **Integration with ResourceSet**
    * View Set
    * XA Transaction
15. **Each Repository Once per ViewSet**
    * View Set Complains
16. **Multiple Views per Session**
    * View
    * Audit
    * Transaction
    * View Set
    * Each Repository Once per ViewSet
17. **Scalability through Revisions**
    * Object in the View
    * CDOID
    * Revision in the Session
    * Reference Targets also have CDOIDs
    * Target CDOIDs Stored in Revision
    * Remove Reference
    * New Revision with removed CDOID (on Commit)
    * Change Reference
    * New Revision with changed CDOID (on Commit)
18. **Views are Revision Selectors**
    * Standard Views are Read-Only and show Latest State
    * Audit Views are using the same Object Cache
    * But select Historical Revisions based on Time Stamp
    * Transactions are using the same Object Cache
    * But manage Local Revisions for Dirty Objects
19. **Object State Machine**



1. **Typical CDO Server**
   * Session Management
   * Package Management
   * Revision Caching
   * Storage Adapter
   * Connects to Storage Backend
   * Hibernate Adapter
   * Uses Hibernate for RDB Communication
   * Objectivity Adapter
   * Communicates with ODB
   * TCPAcceptor
   * Accepts Connections through TCP Socket
   * HTTPAcceptor
   * Accepts Connections through Web Server
   * JVMAcceptor accepts In-Process Connections
2. **Live Demonstration**