LinuxWorld NetworkWorld Conference & Expo
Metro Toronto Convention Centre, Canada

Smalltalk Solutions Toronto April 2006

Translating Smalltalk to Java:

The Good, the Bad and the Unbelievably Ugly

April 25, 2006

S238

Karen Hope

www.beckEngineeringLLC.com

beck engineering

Who Am I ?

- B.S. in Computer Science and M.S. in Operations Research
- Started working in Smalltalk in 1992
  - VisualWorks, VisualSmalltalk Enterprise, VAST
- Worked continually in Smalltalk for various companies until 2004
- Major insurance company 1994-2004
  - Frameworks team lead
  - Process development team lead
  - Trainer
  - Design consultant lead
  - Application Architect
- Currently independent consultant

Application Pedigree

- Insurance policy Writing System enjoyed following achievements:
  - 1997, Smithsonian Innovator Award
  - 2001, 2002, 2003, ACORD Early Technology Adopters, Business Process Reengineering Award, Trading Partners Award
  - 2002, Insurance Journal, Feature Article
  - 2002, Insurance Networking News, Feature Article
  - 2003, Patent Application Filed
  - 2003, Finalist, Innovator Award ***Application Development Trends*** magazine

Audience: Please Adjust Your Expectations

- No code examples
- No conversion of UI to Java
- No companies will be bashed or recommended
- Any information about migration companies is somewhat out-of-date (circa 2004)
- No financial information will be presented
  - Large companies, deep pockets for IS
- Recurrent theme: Smalltalk *good*, Java *bad*

References: Smalltalk Vendors, Migration Services

- Dolphin
  - http://www.object-arts.com/content/navigation/home.html
- Instantiations (VAST)
  - http://www.instantiations.com/
- Knowledge Systems, Inc.
  - http://www.missionsoft.com/
- Squeak
  - http://www.squeak.org/
- Synchrony Systems, Inc.
  - http://www.sync-sys.com/
- Cincom (VisualWorks)
  - http://www.cincomsmalltalk.com/

Application Origin

- 'BFS' == 'Business Foundation System' == 'Foundation' used for Small Business Owners
- Quote, Rate, Endorse, full Workflow automation
- Originally started as rich-client VSE application
  - Deployed in 1996
  - Single state, single product
  - Final architecture deployment in April 2004
- Eventually, multi-line, multi-product, multi-language, multi-dialect implementation (i.e., Smalltalk, XML, Java, Websphere, DHTML,RMI, JSPs, Javascript)

Application Statistics

- Over 2000 classes
  - Domain, Frameworks, Adaptors, Proxies, etc.
  - Wholly deployed on client (other than DB) using notebooks
  - TOPLink for O/R mapping
  - MVC, home-grown UI frameworks
  - Full life cycle support including workflow management
- Internal users, company agents as well as independent agents
- Over $600K active policies in force
- Early adopter of ACORD XML to facilitate comparative pricing of quotes in 3rd party system

Application Architecture Statistics

- Had 5 Smalltalk multi-processor servers
  - 6 - Edits/Rating Smalltalk image clones
  - 4 - Domain service Smalltalk image clones
  - 2 - XML translator Smalltalk image clones
- 2 Additional Smalltalk multiprocessor servers
  - 2 - Downstream/Extractor Smalltalk clones
- Bootstrap Java code NT service managed Smalltalk clones on Smalltalk servers
- Each image ran as an independent Windows Process
- Each image had independent caching strategy
- Initially, round robin dispatching
  - Switched to more discretionary load balancing policy due to uncommon but active HUGE policies

Application Behavior Statistics

- Average insurance policy had 1 building or 4 vehicles
  - However, 3-sigma policies with 140 vehicles, 70 buildings
- Average 200 rates/hour
- System availability 20x7 (over 99% uptime)
- Average real-time edit/rate took 8 seconds
- Over 40 production code base releases during lifespan (1996 - 2006)
- Typically 75 of programmers
- Had 50 on-site production support, help desk, design task force, actuaries and underwriters

What Was at Stake in 1996?

- Total redirection of I/S resources
  - Two of sixty developers knew Smalltalk, OOA&D
  - Retrained Cobol, DB2 programmers
  - Hired college grads, few were Information Systems, fewer still Computer Science
- Started with single product, single state
- Unprecedented partnership – I/S & Business
  - OOA&D natural mutual language
- Began building library of regression test cases using WinRunner

Why Smalltalk (Part I)?

- Originally chosen by Application Architect in 1993
- Future direction of I/S unclear – Windows not certainty
- Smalltalk delivered best environment for cross-platform deployment
- Future of Object Oriented methodology very promising

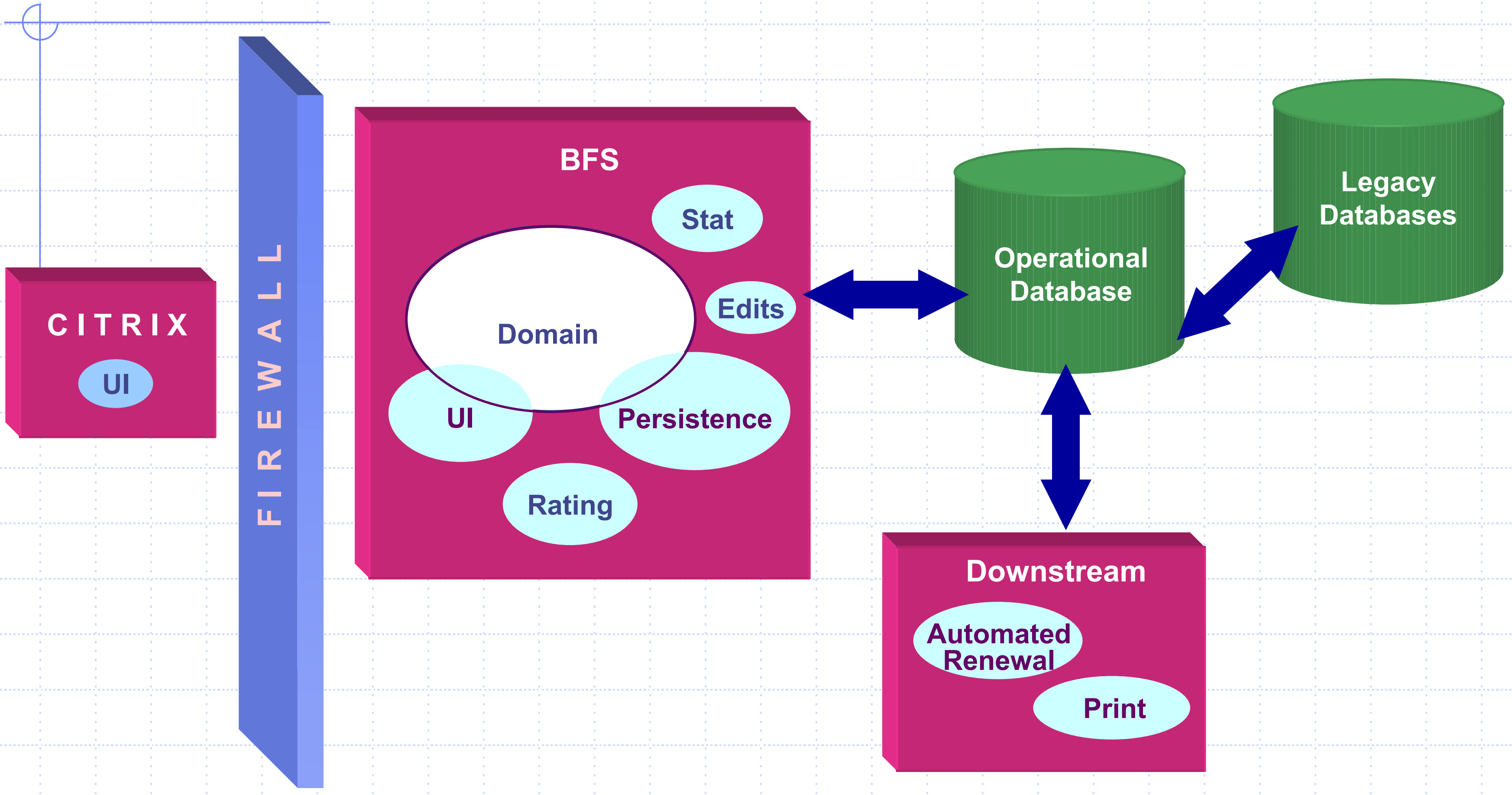
Why Continue Smalltalk (1997 onward)?

- Huge investment
  - Engineering talent
  - Building, modifying application
  - Training, retraining personnel
  - Business commitment
  - Processes

Why Continue Smalltalk (continued)?

- Quality
- Naturalness of language facilitated unprecedented dialog between business and I/S
- Flexibility of language, design facilitated quick release cycle
- Team development was process-driven, became 2nd nature to organization

BFS Architecture 1996-1998



Why <u>Not</u> Smalltalk?

- In 1998, company was purchased by larger insurance company
- Redundant systems
- Months, years of political infighting over which system would prevail
- Smalltalk vilified as "weak link" by corporate

Reasons for Refactoring, Retooling

- App growing too big
  - Too slow to launch
  - Unreasonable memory requirements for users
- Distribution headaches
- Production management complexity
- Needed to broaden our user base and make lightweight quoting available via web
- Serverization seemed natural progression

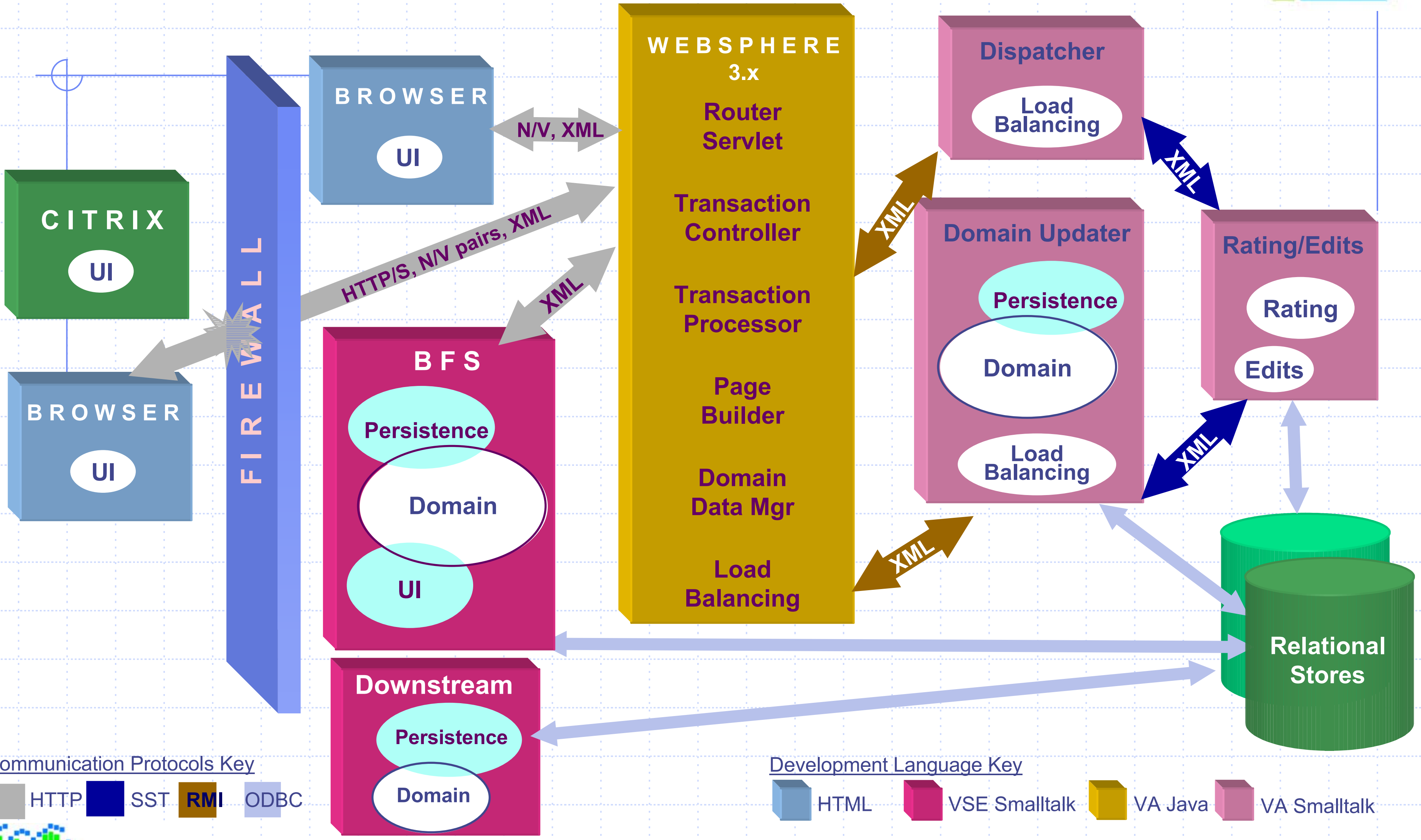
Phase 1 : Introduction of SOA 1999 - 2001

- Further separated logical subsystems
  - Removed “edits” from domain
  - Table/formula driven, similar to rating subsystem
  - Business partners created and maintained formulas
  - Rating/edits changes no longer required coding change, merely data release
- Further reduced dependency on domain for post-processing
  - Created replicated database with 3rd normal form for data
  - Introduced MQ for downstream Smalltalk systems
    - Automated renewals
    - Agency download
    - Stat feed

Phase 1 : SOA 1999-2001 (continued)

- Re-tightened implementation of MVC
  - Some laxity of original design had crept in
- Simplified O/R mappings for post-processing subsystems
- New database design simplified retrieval and instantiation of objects
- Reduced demand on OLTP, gaining real-time performance benefits, reducing deadly embrace, etc.
- Asynchronous processing freed users
- Edits/Rating to operate on DOM created from XML

BFS Architecture 2000



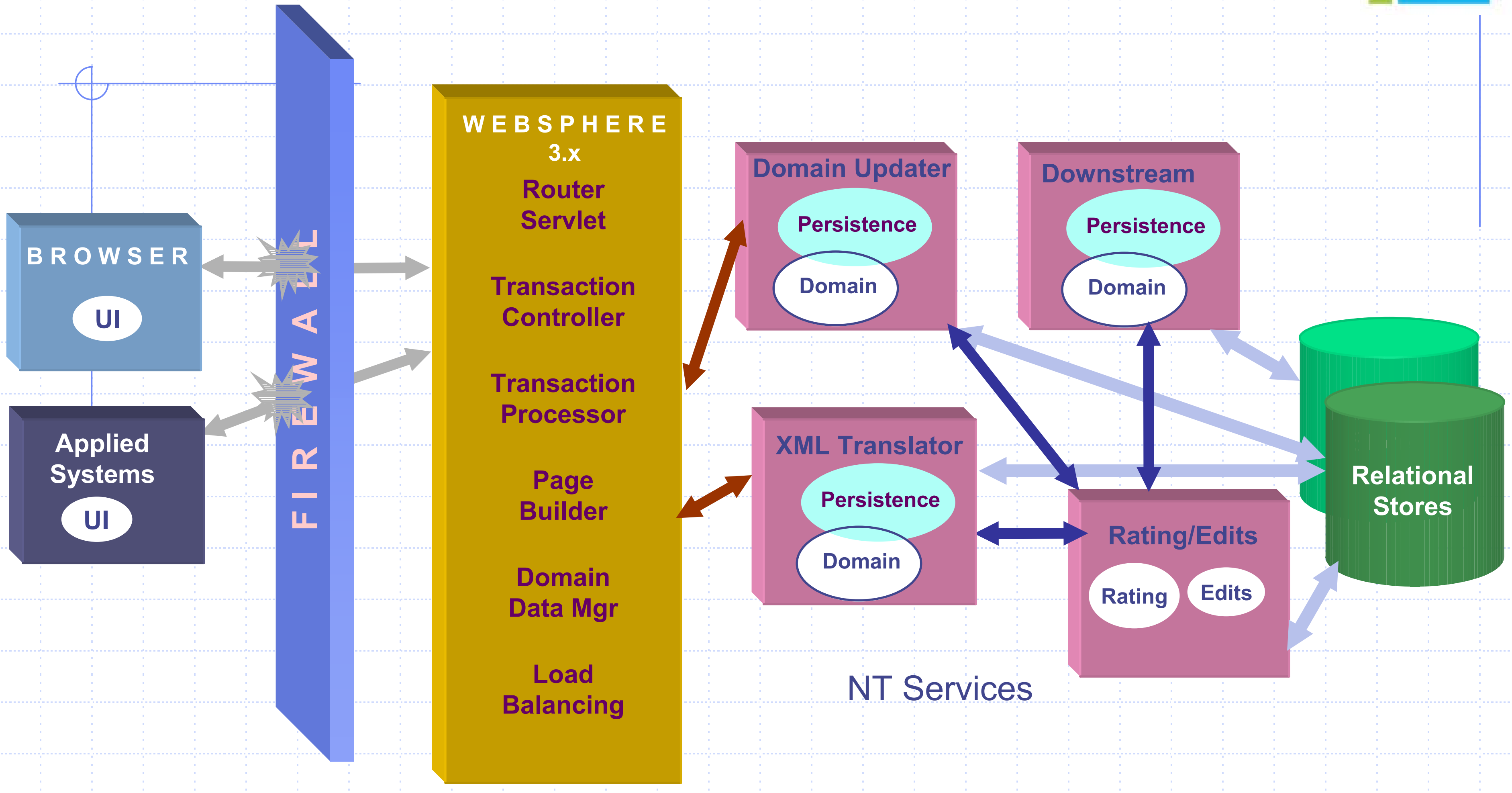
Communication Protocols Key

⬜ HTTP ⬛ SST 🟫 RMI ⬜ ODBC

Development Language Key

🟦 HTML 🟥 VSE Smalltalk 🟨 VA Java 🟪 VA Smalltalk

BFS Architecture 2002



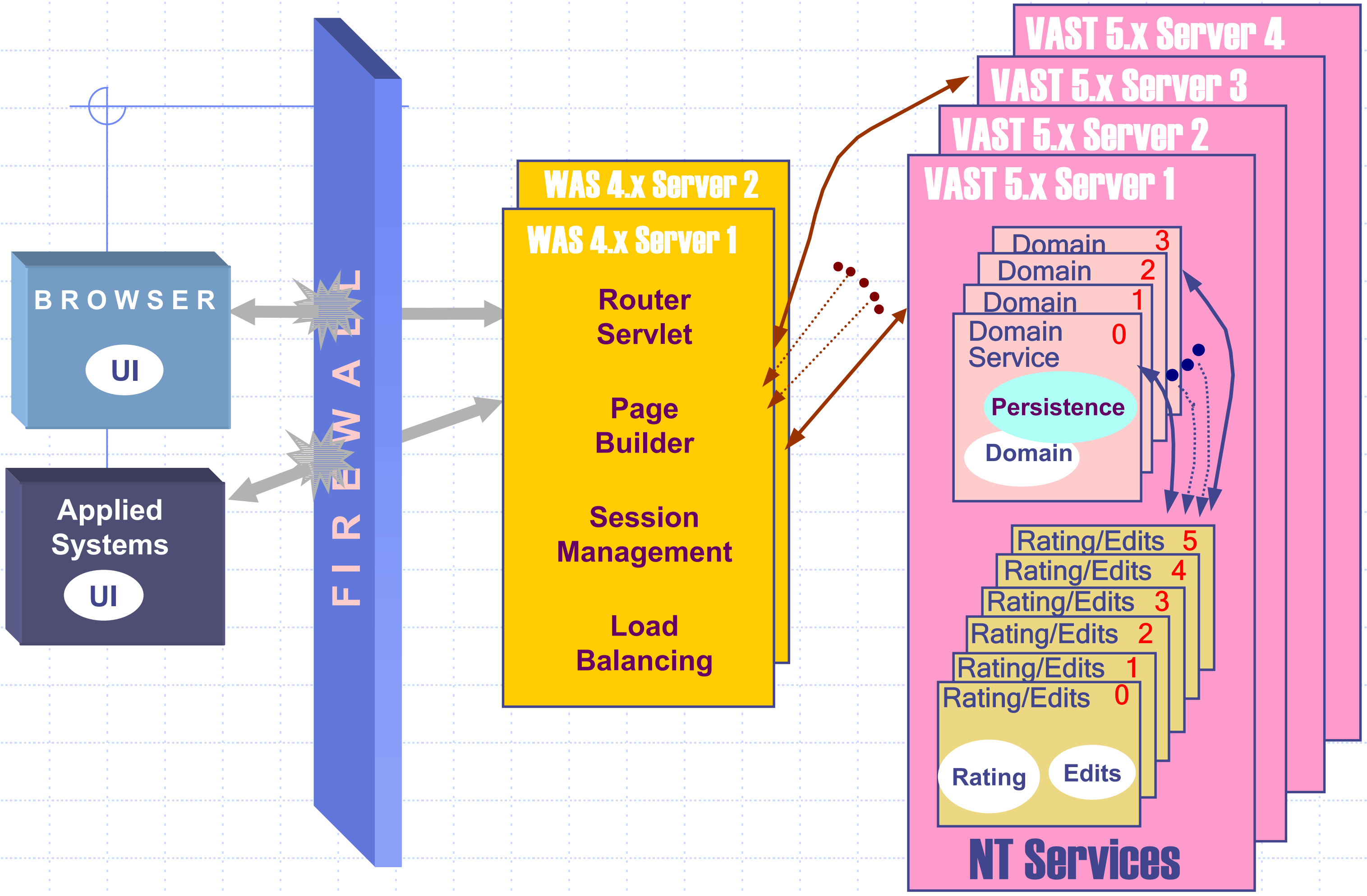
Communication Protocols Key

⬜ HTTP 🟫 RMI XML 🟦 SST XML ⬜ ODBC

Development Language Key

🟦 HTML 🟨 VA Java 🟪 VAST ⬛ External

BFS Server Architecture 2002



Servers Not Shown:
XML Translator
Downstream

Communication Protocols Key

▨ HTTP ▨ RMI XML ▨ SST XML ▨ ODBC

Development Language Key

▨ HTML ▨ VA Java ▨ VAST ▨ External

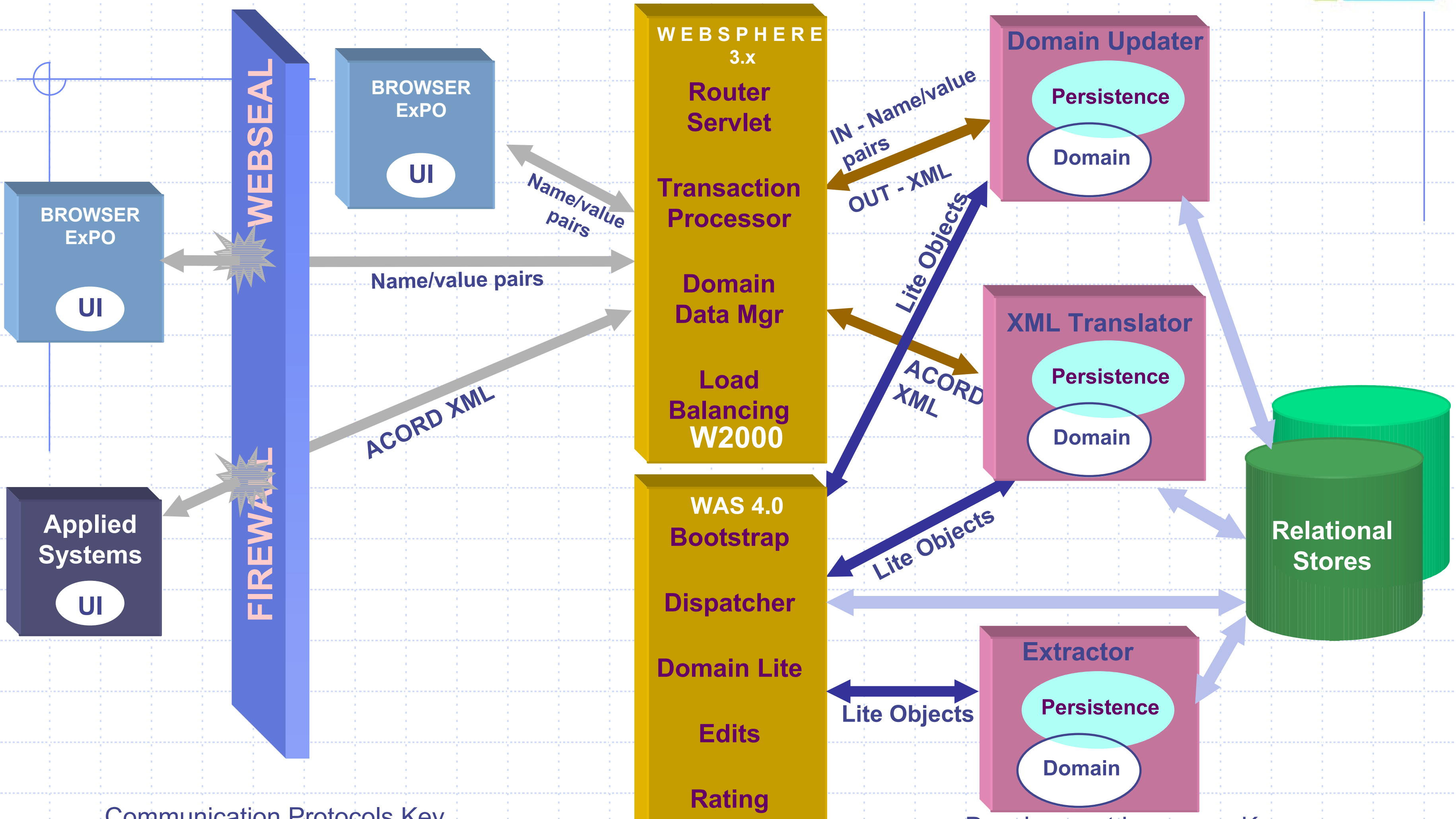
Phase 2: Translation #1 (2001-2002)

- American company specializing in translation
- "Throw it over the wall" approach
- Application team delivered code base and 600 test cases (XML input, expected results)
- Application team refined SOA to use Java Servlets, Websphere
- Project canceled –
  - AFTER VisualAge Java Rating and Edits ran correctly!!
  - Only reasonably late
  - VisualAge Java code was able to reproduce correct results in over 600 test case
    - (In fact, found a few bugs in our test cases)
  - Gave opportunity to critique code, leading to iterative re-engineering

Phase 3: Translation #2 (2003-2004)

- American company specializing in migration
- Hands-on approach
  - Application engineering staff had used their companion product for several years as part of development cycle
- 6-20 application engineers
  - 6 Full-time
  - 20 During heavy regression/load testing
- Only reasonably late

BFS Architecture 2004 - 2006



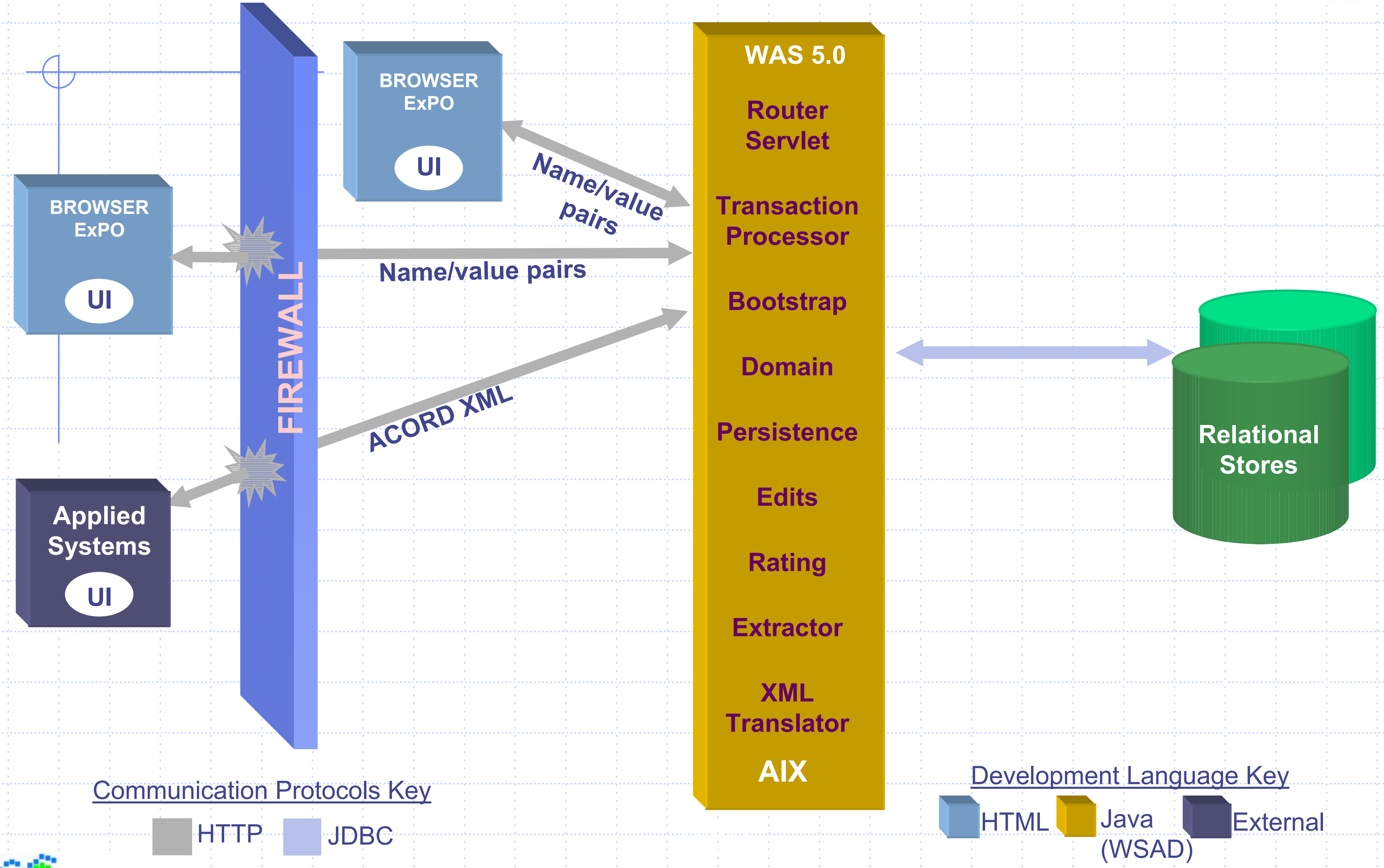
Communication Protocols Key

▨ HTTP ▨ RMI Lite Objects, XML ▨ JDBC, ODBC

Development Language Key

▨ HTML ▨ Java WSAD ▨ VAST ▨ External

BFS Architecture 2006 (Proposed)



Summary of 3rd Party Experiences

- Both consulting companies succeeded
- Both consulting companies had problems
- First HUGE code base for both
- Application was “bleeding” edge in both cases
- Funded R&D for both

Current Status 2006

- Company once again acquired by another insurance company in 2002
- Once again, redundant systems
- Decision made to go with other system
- BFS sunsetted, or soon to be

References: Java vs. Smalltalk Contrasts

- Bothner, Per:
  - http://per.bothner.com/papers/smalltalk.html, 1996
- Boyd, Nick:
  - http://www.educery.com/papers/sttojava/, 1997
- Davis, Ryan:
  - http://www.zenspider.com/Languages/Smalltalk/VsJava.htm, 1997-2004
- Fusselll, Mark:
  - http://www.chimu.com/, 1997-2000
- Giorgi , Giovanni:
  - http://daitangio.homeip.net/squeak/squeak_tutorial.html, 2002
- Laffra, Chris:
  - http://www.developer.com/tech/article.php/614371, 1999(?)
- Logan, Patrick:
  - http://www.whysmalltalk.com/articles/pages/javavssmalltalkblocks.htm, 2002-2005
- Raab, Don:
  - http://www.whysmalltalk.com/articles/raab/productivity.htm, 2002-2005
- Ross, Niall:
  - http://wiki.cs.uiuc.edu/CampSmalltalk/Smalltalk+for+Java+Programmers

Smalltalk Syntax, Idioms or Paradigms Awkward to Translate

- Types
- Blocks
- Thorough use of Class behavior, Class Instance variables
- Calculations (rounding errors)
- Dates suck

Smalltalk Syntax, Idioms or Paradigms (continued)

- Multiple return object types
- Primitive types or wrappers ?
- Special behavior coded for DNU
- Objects inheriting from *nil*
- Use of #perform:* obscured types
- Formula translation difficult to type
- Needed to write "utility" or helper classes

Smalltalk Syntax, Idioms or Paradigms (continued)

- Debugging marshaling errors
- Casting errors, recasting errors
- RMI slow
- Had to write Java classes to duplicate Smalltalk class behavior
- Maintaining dual systems (Smalltalk domain, Java Lite) problematic
- Merging, integration, build processes complicated
- “Errors” due to bugs in test cases, increased rounding errors

Specious Reasons for Translation

- Future has no shadow
- More people know Java
  - shorter learning curves
  - more cheap, available programmers
  - reduced training needs
- Code will be more maintainable
- Deployment will be more robust
- Java, Eclipse are free!
- Application will now use state-of-the-art technology and architecture
- *Translation* means no additional Smalltalk work

Realized Benefits

- Performance!
- Average rate went from 8 seconds to 2 secs
  - Caveat: New caching strategies contributed to increased performance
- Abandoned J2EE component (shared memory) – too slow
  - Legacy from Smalltalk
  - System performed just as well without it

Realized Benefits (continued)

- Reduced number of servers
- System availability increased
- Reduced cycle time for automated renewals
- Reduced maintenance training needs
- Happy executives
- Resigned, but happy, business partners

Conclusions

- Be sure you know why you're doing it
- Use combination of experienced and junior personnel
- Be ruthless!
  - Need clean code with independent configuration maps (or equivalent)
  - Get rid of dead code
  - Rewrite blocks where possible
- Cute, clever, pithy Smalltalk code can lead to dreadful Java code
- If only a few people grok Smalltalk code, fewer still will understand or want to maintain said Java code
- Crappy Smalltalk code stinks ten times worse in Java -- and there's more of it

20-20 Hindsight Conclusions

- Rewriting entire module from the beginning would have been cheaper, and just as much fun
- Extremely lucky
  - no major persistence issues
  - no UI
- Possibly would have used Web Services
- Possibly should have migrated to .NET
- Nature of software – nothing stays the same, no company can stand still
- Gotta come to work anyway…

LINUXWORLD. NetworkWorld

CONFERENCE & EXPO

Metro Toronto Convention Centre, Canada

SMALLTALK SOLUTIONS TORONTO APRIL 2006

Translating Smalltalk to Java:

The Good, the Bad and the Unbelievably Ugly

April 25, 2006

S238

beck engineering

Karen Hope

www.beckEngineeringLLC.com