

Hey You! Get On To My Cloud!

Application Development and Deployment in the Cloud



Dave Thomas
Bedarra Research Labs, Object Mentor
Queensland University of Technology,
Carleton University
JAOO Australia Conferences
www.jaoo.com.au
www.davethomas.net
dave@bedarra.com



Application Development for the Cloud



The Challenge

- Middleware – Life in The Tar Bit
- Next Generation Application Challenges
- Business Agility – The Driver for Change
- Lean and Agile – Towards Agility
- BUT !?!

The Solution

- Next Generation IT
- Pervasive Computing Power – The Hardware Enabler
- Infrastructure, Management and Development in the Cloud
- Cloud Side Application Development
- Client Side Application Development
- Do It Yourself Cloud Programming

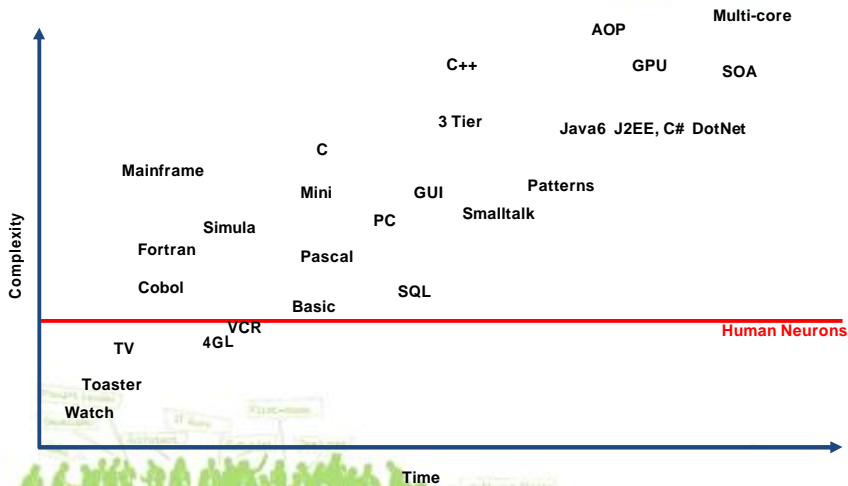


Jurassic Middleware: Life in The Tarbits



1. Prehistoric Software As A Service – Mainframe Dinosaurs, Simple Verbose Cobol, PL/I, Fortran and Cryptic but powerful APL
2. The Age of Complexity – Client-Server, C/C++, SQL PSQL, TSQL...
3. The Age of Naïve Simplicity – Spreadsheets and Enterprise Applications
4. The Age of Absurd Complexity – Objects, Middleware, XML, BPM, SOA...
5. Life after Middleware (Post-Modern Computing) – Cloud Computing?

Complexity of Technology



The Daunting Difficulty of Application Development

JAOO

- ↑ API Surface Area = API x Frameworks
- ↑ Language Surface Area = Grammar Productions x Languages
- ↑ Ways of Doing The Same Thing = Platforms x (2 to 4)
- ↑ API Stability = (Middleware + Upperware + Lowerware) x 3 versions
- ↑ Accidental Complexity
- ↑ Developer IDE Features = Editor + Browser + Build & Test + Versioning + Process + Models x (1-3)
- ↑ Klocs Per App Delivered
- ↑ % of Budget for Maintenance vs. New Development
- ↓ Readability of Code
- ↓ Locality of Application Code
- ↑ Developer Certification versus Competence
- ↑ Global Shortage of IT Skills
- ↑ Vendors Say Life is Getting Better



The Escalating Costs of Ownership

JAOO

- ↑ Software Tiers
- ↑ Hardware Tiers
- ↑ Software Stack
- ↑ Software Tools – IDEs, Build, SCM, SQA, Modeling, Performance
- ↑ Installation and Upgrades
- ↑ Vendor Interoperability
- ↑ Enterprise Open Source Version Management
- ↑ Recruiting and Retaining Top Talent
- ↑ Development Maintenance as a % of IT Budget
- ↑ Platform, Framework, Tool churn rate
- ↑ License Complexity and Costs
- ↑ Vendor Lock
- ↑ Vendors say things are getting better



The Challenges of Next Generation Applications

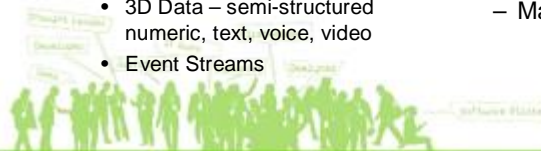


Real-time Business

- Development in Real-time
- Execution in Real-time
- Deployment in Real-time (zero install)
- Global Applications Deployed Directly To Any Device
- Massive Amounts Of Data
 - Raw Processing versus Aggregation
 - 3D Data – semi-structured numeric, text, voice, video
 - Event Streams

Collaborative Applications

- Eliminate artificial technical or organizational barriers
- Cross Functional
- Cross Organizational Virtual Teams
- Selective sharing of data with partners, competitors & customers
- Leverage Service Providers
- Mass Customization



Business Agility: The Driver



- Agility enables the business to respond quickly to customers, partners and competitors
- Agility is a strategic business, hence IT, capability
- Agility implies the ability to rapidly build and configure tailored solutions which span internal and external systems
- Want to exploit the price/performance of new hardware technology
- Agility is defined by companies that your CEO envies:
 - Progressive Insurance
 - Google
 - Amazon, Yahoo
 - Sales Force
 - Facebook, MySpace, LinkedIn
 - Apple



But We Can't Get There From Here?!

JAOO

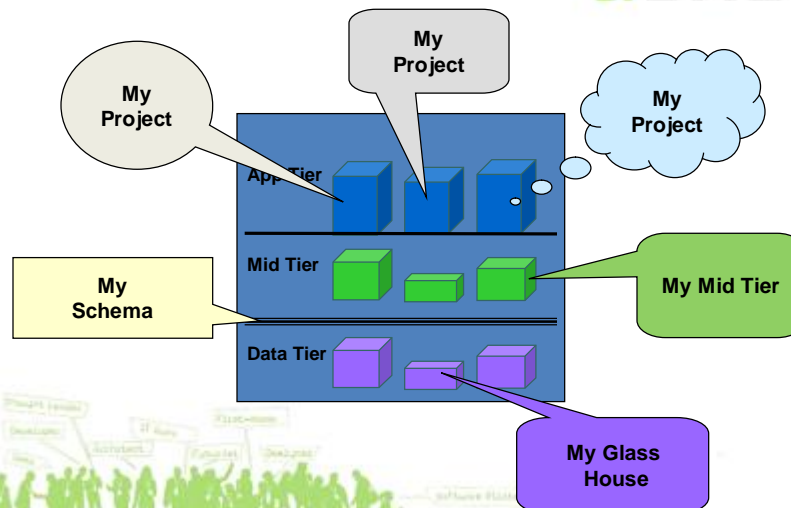
Lean and Agile: Improve predictability and quality, but still we have limited Agility!

- Both Business IT and Software Product Vendors are:
 - Frustrated with their lack of agility in both development and deployment of applications and services.
 - Finding that their best people, equipped with the best practices, tools and middleware can barely keep pace.
 - Concerned with their ability to meet the demands of Next Generation Applications
- Both are coming to realize that the companies they want to emulate are taking a different road with respect to infrastructure, development and delivery which provides them substantially reduced operating costs and increased agility.
- This realization is the primary driver for what we call the Next Generation IT.



Barriers To IT Agility – Techno-cultures and Territories

JAOO



Pervasive Computing Power – The Hardware Enabler



- Processors, Memory, Bandwidth and Storage are “free”
 - 100s to 100000s of processors (“Googleplex”)
 - Oodles of Memory and Gaggles of Disk Storage
 - High Speed Wired and Wireless Bandwidth
 - Multiprocessing, Multi Cores and System On Chip
 - Special purpose processors e.g. GPUs
 - Dynamically Reconfigurable Systems
- Pervasive Connectivity**
- Internet Everywhere
 - Always on but Occasionally Disconnected Devices
- Enhanced Input and Output**
- Audio and Video Input
 - Two handed input
 - Smart Materials
 - Environment, Location Awareness



Global Distributed Data Centers – Tomorrow’s Glass House



“Hey! You! Get On To My Cloud!”*

Google, Amazon, SalesForce, VMWare,
Microsoft Azure and Dell, RackSpace, Akamai, ATT, IBM
Blue Cloud, HP Flexible Computing, Sun Utility Computing

IT Glass House → Distributed Super Computer

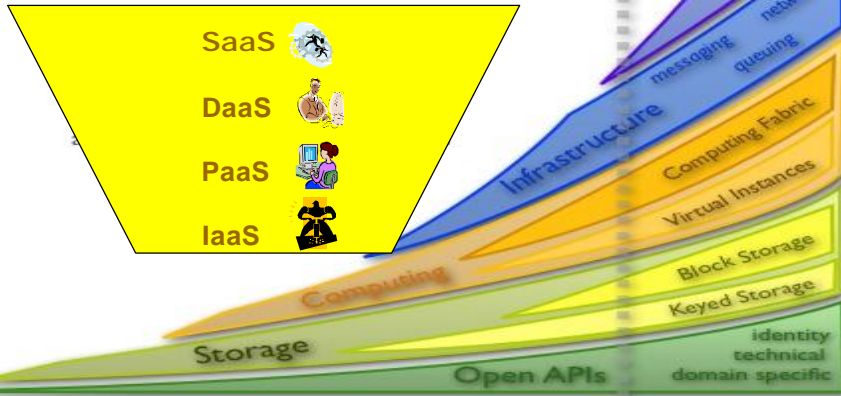
- Lower-cost and lower-power components
- Fault tolerance through failover and replication
- Non-stop operation through upgrades of systems, software and apps
- Embedded security v.s. simple access control
- Automatic computing
- Dynamic resource management



* with apologies to the Rolling Stones

Crossing the Chasm in 2012

The Growth of Cloud Computing



From <http://blogs.zdnet.com/Hinchcliffe>

Estimated Revenue from S3



Exhibit 17

Estimated Revenues From Amazon's S3 Storage Web Service

S3 Usage	1Q:07	2Q:07	3Q:07	4Q:07	1Q:08	2Q:08	3Q:08	4Q:08E
Billions of Objects	5	7	10	12	14	22	29	35
Average Size of Object (KB)	500	500	500	500	500	500	500	500
Usage (TB)	2,384	3,338	4,768	5,722	6,676	10,490	13,828	16,689
Avg Price Per GB (\$)	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
Usage Revenue (\$M)	0.4	0.5	0.7	0.9	1.0	1.6	2.1	2.5

Source: Company reports, Bernstein Analysis



We Don't Own The Infrastructure!



Amazon, RackSpace, RightScale, 3Tera etc.

- Virtualization enabled by VMWare, XTEN, and proprietary hypervisors
- Storage and Data Management
- Processors and Virtual Processors
- Network Bandwidth
- Geographic Distribution – Akamai, ATT, IBM, MS, EC2 Regions
- Scalability and Tolerance
- OS Environments – Linux, Windows, Oracle
- Cloud Network Management - SevOne
- Application Environments – SAP, Oracle ...

Examples

- Target, Marks and Spencer, Second Life, eScience and eEngineering virtual labs



They Manage It!

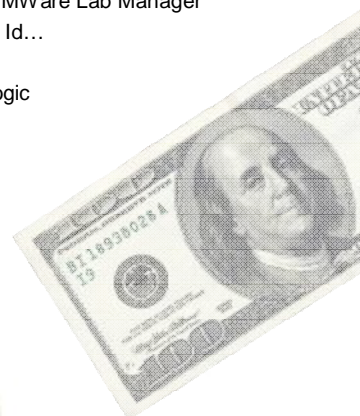


Examples

- Cloud Configuration and Deployment – Apptix, 3Tera and rPath
- Virtual IT Test beds - Skytap, GoGrid, VMLogix, VMWare Lab Manager
- Single Sign On – OpenID, Amazon, Google, MSN Id...
- Cloud Systems Management – SevOne
- Application Integration – CastIron, Boomi, SnapLogic

Follow the money in the cloud ...

- MS Azure
- IBM Blue Cloud
- HP Flexible Computing
- SUN Utility Computing
- ATT
- Akamai



Hey Mate – No Development Environment Here



But I have all the build machines I finally need!

- Cloud Hosted Linux, Windows, Oracle development environments and tool chains
- Virtualization for fast configuration and restore
- Parallel Build and Test and Stress Test
- Source Code Control – Subversion or GIT
- Developer Collaboration – Wiki and Defect Management
- Integration Services
- Examples
 - Atlassian and Cognetix - Jira, Confluence, Bamboo
 - Collabnet- SourceForge and SVN
 - Cloud Hosted IDEs - RAP for Eclipse, more to come ...



Service-Oriented Computing Infrastructure: The Software Enabler



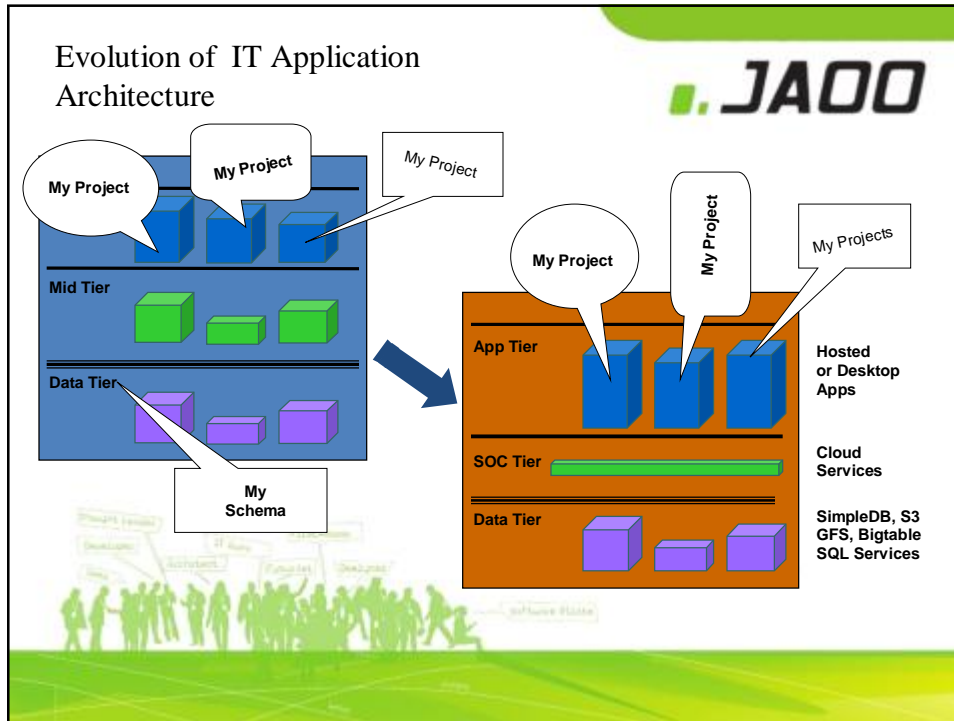
The Emergence of A Simpler Application Infrastructure

- Examples - On Demand, Software As A Service such as Amazon S3, EC2, SimpleDB, Google App Engine, Sales Force ...
- Simpler limited “thin” service API (< 50) closer to underlying platform which provides support for scalable, distributed, secure computing
- Independence on mainstream vendor Underware and Middleware
- Google Linux, VMware Virtual Machine, MS Azure Hypervisor V

Application Development Benefits

- Small Service API (thin to none class library & frameworks)
- Limited Choice Reduces Decisions and Support
- Leverages Other Apps through Services
- Total App Responsibility from envisioning to production
i.e. App Team carries the beeper





Most Applications are Still Essentially CRUD!



- Modern Application = CRUD + Compute + Interact
- CRUD against federated data sources
- Computation against a huge heterogeneous data sets
- Display based on an known information model
- If one ignores objects, heterogeneous data sources and fancy UI, it is essentially still a simple 4GL problem

```
BEGIN MyApplication
```

```

    SELECT what user and/or application needs from
           WHERE it is stored
    THEN
    Perform more filtering and calculations
    THEN
    UPDATE appropriate things WHERE it is needed
    THEN
    Display what is needed

```

```
END MyApplication
```

Cloud Side - Application Development and Delivery

JAOO

- Cloud Side Programming Models
- SOC – Programmable Web/SAAS
- Super CRUD = CRUD + Cloud FP
- Simple and Scalable - ATOM/RSS and REST/JSON
- Applications Scripting
- Cloud Schema-less data bases
- Virtual Application – Instant Deploy
- Designing for Scalability!



Application Experiences using High Order Languages

JAOO

Examples

- Relational Programming – SQL and extensions
- Vector Programming – APL, NIAL, J, K
- Functional Programming – Scheme, Haskell, F#, Erlang
- Set Programming – SETL, Kleisli, XQuery
- Dynamic Object Programming – Smalltalk, CLOS, Ruby
- Data Flow – Pipes and Filters- Unix, Stream Processing
- Logic Programming – Prolog, Datalog
- Constraint Programming – ThingLab
- Reactive Programming – Erlang
- Text Processing – SNOBOL/Icon, Omnimark



Super CRUD Functional Programming for the Masses - The Leverage

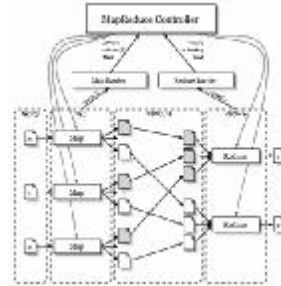
JAOO

Examples

- Google Map Reduce ($f(g(x))$)
- (Apache Hadoop, Connection Machine Lisp)
- Microsoft F#, Azure SQL Services and Dryad

Benefits

- Functional architecture reduces API surface area and enables rapid application development
- Allow developers to think in terms of simple collections independent of their shape and representation
- Enables implicit fault tolerant data parallelism and distribution (i.e. move the function to the data instead of moving the data to the function)
- Transactional Shared Memory enables simpler programming for "state full sinners"



Beyond SQL – Towards a Next Gen Data Language

JAOO

- Collection programming
- Closed operations over tables, sets, dictionaries, lists & arrays
- SQL + higher order functions inspired by Haskell and Vector Languages such as J
- Efficient Deductive Database inspired by Datalog
- Search is cheap and may be lazy
- Computation is close to the data

Unfortunately most developers cannot formulate a complex query in SQL, XQuery, Map Reduce or LINQ... Hence there will be an acute need for developers who can!

Client Side - Application Development and Delivery

JAOO

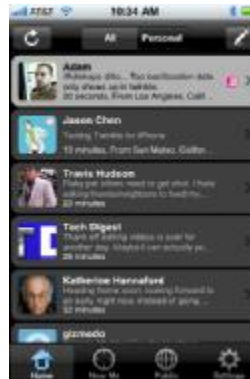
- Client Side Programming Models
- Vector Graphics 2 and 3D
- Multi-Touch, Voice and Interaction
- Scripting – JavaScript, Flex ..., Ruby, IronPython
- Occasionally Disconnected Operation
- Service Assembly and Business Programming
- Security and Privacy



Real Men Don't Do JavaScript!?

JAOO

- Use the right technology for the right job
 - Scripting for application, SOC for services
- JS is the new BASIC/Smalltalk/Scheme!
 - JS is pervasive and good enough
 - Children can do it
 - Normal developers can do it
- For example Apple iPhone JS enables simple access to Cocoa and multi-touch UI allows compelling apps without the pain of Objective-C and Cocoa

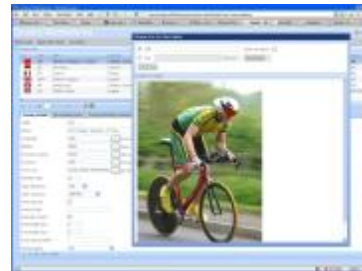


Real Men Don't Do JavaScript!?



Why JS now?

- JS is pervasive
- JS is good enough
- JS has prototypes and higher order functions
- JS libraries are maturing e.g. You, Dojo, JQuery and EXT JS ...
- Higher quality, increased speed and space efficient runtimes are available
- IDE support is improving



Google V8 is open-source JS+C++

- Provides open source VM for everyone
- Smalltalk like runtime
- Full C++ Interop
- Eliminates license hassles

Need a Flashy Active Non Web UI



- Flex and Air enable unique interaction experiences
- Take a look at the early work on Lively - JS + Squeak Morphic + SVG e.g. Ward's Piston and Weather - Look Ma - No Darn Tags here!
- See also the O3D project



The Future of MS Client Side Applications

JAOO

- MS is already leading the back with XAML and WPF which provides a vector based UI , scripting and execution platform for the desktop
- Silverlight and the new Expression tool chain enable rich client development and delivery across platforms
- MS is in the race with Google, TraceMonkey, Squirrelfish to provide a high performance JS.
- F# language and Dryad research (LINQ for Cloud) provide cloud programming capabilities
- While still speculative MS Volta allows the CLR code to run in any browser



Occasionally Disconnected Applications

JAOO

- GOOD News - The Internet is almost pervasively available
- BAD News - Applications are occasionally, sometimes frequently disconnected.
- Applications need to be designed to accommodate disconnected operation
- REST-full design allows client to maintain state via a "continuation"
- Google Gears – Program against a local web server which is prepared to handle synchronization when it reconnects. Advantage is that application can still be a web app.
- Microsoft Live Mesh – Powerful Synchronization Framework supports a wide variety of client – cloud synchronization
- Workflow based transactions versus classic atomic transactions



Security and Privacy



- For many SMB clouds are likely more robust and secure than their current environment.
- Best practice is to encrypt your data using a client or server side encryption service (could be in the cloud).
- SSL V2 is essential
- Improved Browsers and JS tooling such as Caja improve Ajax however much remains to be done (cross site script loading)
- Statefull reverse proxy can provide a simple means to secure stateless REST services
- Standard audit techniques such as dual entry and checksums should be used for additional confidence.
- Biometrics and dynamically generated challenges reduce fraud
- Leverage other trusted services Secure Service - Credit Card processing services have strong fraud detection as part of their services



Democratizing Cloud Application Development



- Trust the Web Luke! REST, JSON, RSS and ATOM enable loosely coupled access to cloud and legacy data
- Support simple higher order patterns such as map reduce, tuple spaces and actors to leverage cloud resources
- APP provides a simple useable CRUD collection API
- Dynamic RAD Development is Back! Ruby, Groovy, Python allow rapid development of web apps on Linux, MS.NET or Java infrastructures
- JavaScript is the next generation Web 2.0 application development technology
- End User DSLs can further accelerate application development by leveraging well know workplace tools and domain knowledge – e.g. Enterprise Mashups
- Office applications can grow into the Cloud e.g. Spreadsheets, desktop data base ...



Do It Ourselves Programming



The Empowerment Domain Oriented Programming and DSLs e.g. 35 million second life scripts

Business Driven Development

- Business teams with domain experts and developers embedded in the team
- Use of Domain Specific Language
- Scripting, Assembly and By Demonstration Programming
- Leverage Fine Grained SAAS offerings
 - FireApps, Google, Yahoo, SAAS providers
 - Programmable Web
 - Legacy exposed ATOM feeds



Do It Ourselves Programming –The Empowerment



Examples of Business Programming by Knowledgeable Business Users

- 4GLs – Synon, Natural, Mapper, ZIM, Cool Gen, Access
- Programming By Example – QBE/OBE/SBA, Tinker
- Rule Programming – Business Rules, Expert Systems, Decision Tables, State Tables, StageCast
- Spreadsheets – Excel, The CIA Analyst, AgentSheets, DabbleDB
- Mathematical Programming - Matlab, Mathematica, Maple
- Visual Languages – Prograph, Labview
- Pattern Matching – AWK, Perl



Do It Ourselves Programming –The Empowerment

JAOO

Business Driven Development

- Enterprise Mashups – The Real SOA?
- Applications Assembled from Feeds and Services



IBM QEDWiki



Yahoo Pipes



Dabble DB



Lively Fabrik

The Future Weather is Cloudy With Lots of Sunny Spots for Applications

JAOO

Cloud Computing Enabled Domain Oriented Programming

- Simple Services enable Agility and Leverage Scaleable Commodity Technology
- Functional Services enable rapid application development and enable the service infrastructure to handle concurrency
- JavaScript and associated tooling democratize application development
- Domain Oriented Programming enable domain specific service development
- Business Programming enables business teams with embedded developers to deliver applications



Thank You!

