WHITE PAPER

Enterprise Software Systems at Web-scale

Extending the Useful Life of Software for Lasting Value.

Akashi Kaikyō Bridge, source: *famouswonders.com*

Executive Summary

In order to better align enterprise software systems with business and protect these investments in the face of future changes, corporate IT departments must embrace Web-scale in software engineering including:

- economies of scale
- longevity
- level of abstraction

Developing Web-scale, not just object- or service scale, software will increase their ability to re-use resources and re-align IT to future changes in business; extend the useful life of their systems and enhance their value; reduce their computing footprint in the cloud and reduce total cost of ownership.

"By 2017, Web-scale IT — a pattern of computing that delivers the capabilities of large cloud service providers within an enterprise IT setting — will be an architectural approach found operating in 50 percent of global enterprises", according to **Gartner Inc.**

That estimate is a significant increase from less than 10 percent in 2013.[1]

1 *Gartner: Global Enterprises Increasingly Adopt Web-scale IT*. Bob Violino, Information Week. March 6. 2014. http://bit.do/Web-scale-IT

Enterprise Software Systems at Web-scale

What is Web-scale?

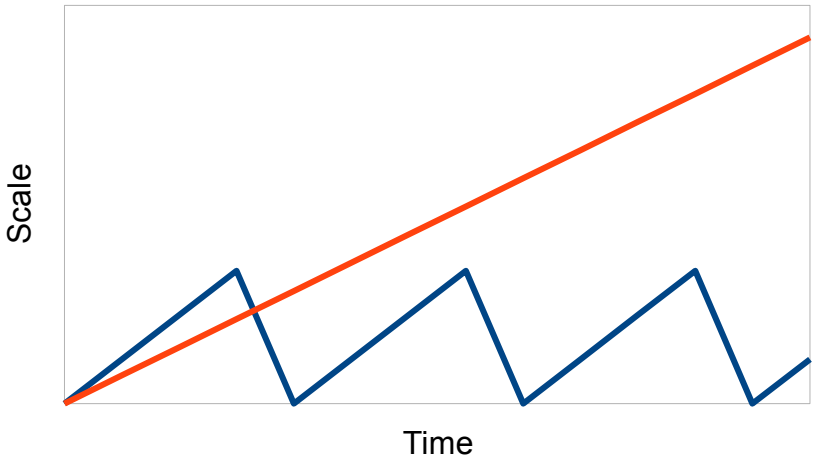
Building enterprise software systems with characteristics of the Web means three things:

1. Economies of scale, achieving its level of automation and availability
2. Longevity, achieving its flexibility in adapting to change over decades
3. Abstraction, using Web-orientated architectures to evolve

Scale. Any department running an application in the Cloud, public or private, stands on the shoulders of IT giants and benefits from their geographic reach and operational excellence in providing connectivity, managing storage and perfecting data center design. Many corporations already take advantage of these economies of scale and in future will be able to buy or sell capacity in commodities markets[2].

Longevity, a property so obvious it is easy to miss. The Web has been running without a reboot and adapting to continuous changes since March 25th 1989; its useful life, as of this writing, is 25 years and counting.[3] The longevity of enterprise software systems does not nearly approach Web-scale. Software of sufficient complexity inevitably becomes brittle, accrues technical debt and reaches a state of un-maintainability. At exactly that point in time, when maintenance costs equal the cost of rewriting applications and their integrations, entire systems have to be replaced.

Economics: Software Saw-tooth vs. Web Evolution



2 *Deutsche Börse to trade cloud services*. Financial Times, July 2nd 2013. http://on.ft.com/ODmOo0
3 The Web Turns 25. PBS.com, Feb 27th 2014. http://to.pbs.org/1dFhWEn

This happens every 4-6 years, which is the shelf-life of enterprise software systems even today. If managers want to deploy enterprise solutions with longevity at Web-scale they must first pinpoint options that extend their useful life at least three fold. These managers are not well served by the market. Both currently dominant IT trends, agile/lean software development practices and the DevOps movement, define ways to reduce the cost of writing and deploying software but neither of them has the ambition to extend its life span by an order of magnitude.

Until industry brings into focus and finds a way to create evolvable architectures[4] that extend the useful life of enterprise IT, the economics of software development and operations can not reach Web-scale. The short lifespan carries a cost: IT departments will be forced to repeat the saw-tooth-pattern of rebuilding and reinvesting every 4-6 years despite outsourcing development to offshore centers and moving applications to the Cloud. Moreover, this pattern makes it very difficult to align IT with business strategy and protect its value. A high turn-over rate in the code base can be just as constraining as a high turn-over rate in human resources.

"In evolution, you never build something new if you can adapt something you've already got."

David Linden, professor of neuroscience Johns Hopkins University

Abstraction. Getting "abstraction" right–learning to choose between a tele- and microscope–is required to achieve Web-scale IT. There are three elements in the Web abstraction: the "world" maps unto a space ("WWW") populated with uniform resources (URLs) that interact (REST). In contrast, enterprise software systems abstract at the level of objects- or services. These lack uniformity, do not populate any space around them and consequently lack the quality and structural integrity required to manage very large complex systems. The object- or service abstraction[5] has proven incapable of organizing the complexities of the Web connecting billions of endpoints, let alone in an economically viable way.

If corporations really want to benefit from Web-oriented architectures and Web-scale IT, then enterprise architects must embrace its abstraction based on concepts of

4 SOA cannot deliver on this promise because it too is a static and not a dynamic paradigm.
5 As in Object-Oriented Programming and Service-Oriented Architecture

space, uniform resources and decoupled interactions.

If there were a secret formula for Web-scale IT, this abstraction is it. It reduces the costs associated with software development and the costs associated with the integration of complex systems; it also lowers the cost of making changes in architecture at any time–uniform resources can be easily changed, added or constrained. The outcome is that a business architecture can evolve through countless iterations performed by Agile teams over long periods of time without hitting that wall of un-maintainability native to traditional enterprise solutions.

Adapting Enterprise Software Systems to Business

At Web-scale, in short, denotes a long-lived and evolvable software system. What makes Web-scale so attractive from a business perspective, aside from cost savings? Well, the Web provides a tried and tested model for a large complex system that adapts over long periods of time to changes in its environment. To protect investments in software systems in the long term, CIOs not only have to better align IT to current business goals, but to be able re-align IT and software to yet unknown and new goals in the future, because the business environment always changes. To that extent resource-oriented software systems at Web-scale are the digital platform that enables companies to imitate the Web and CIOs to adroitly align and re-align IT with business.

What is the best strategy to re-align IT to future changes in business? A strategy we have been using for millennia while sparring with the invisible forces of natural selection provides the answer: “In evolution, you never build something new if you can adapt something you've already got.[6]”

In other words, the ability to align IT to business and re-aligning it to future changes hinges on the rate of re-use inside of a software system. In this context, “rate of re-use” is one of the *key performance indicators* by which to measure Web-scale enterprise software systems in terms of their longevity and evolvability. Web-scale, specifically uniform resources, allow for a rate of re-use significantly higher than anything in the object- and service-oriented software world.[7]

The high re-use rate directly impacts project budgeting. At below Web-scale, at the level of object- and service orientation, total cost of ownership of a system is split 30:70 be-

6 David Linden, professor of neuroscience Johns Hopkins University

7 Software built at resource-oriented Web-scale retires the practice of data binding in order to keep information separate from code at all times.

tween Devs and Ops[8]. The lion share of any enterprise application budget is used up after Release 1.0. It is spent to maintain applications, make necessary changes in existing code so new features can be added and to counter software entropy. At Web-scale this expense ratio between Devs and Ops turns upside down and changes from 30:70 to 70:30.

Re-use of resources not only impacts the expense ratio, it also effects gains in productivity and indirectly lowers total cost of ownership. One can observe that already during development and systems integration. In one instance a project manager reported a 20-fold drop in man-hours during one of the largest multi-national systems integration projects of its kind.

Low cost of change and high rate of re-use, these are the properties an architecture must provide to embody Web-scale. Together they ensure the longevity of an architecture and allow enterprise software systems to evolve.

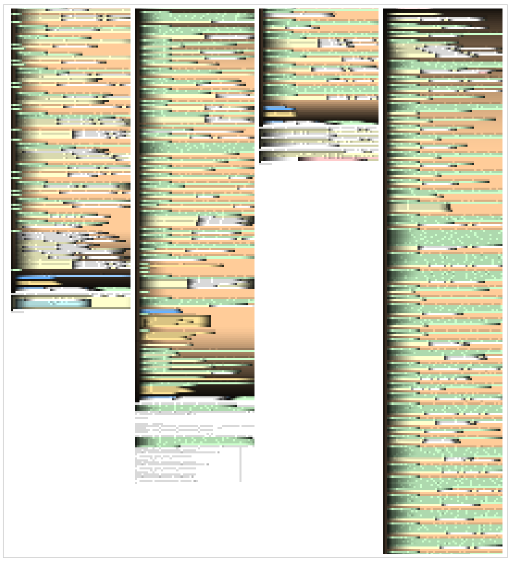
Not everybody agrees that Web-scale is, by definition, good for business. A criticism leveled against Web-scale IT is that not every business is as big as Facebook, Google or Amazon. Also, for most companies IT is a means, not an end, and they will always want to be consumers not producers of IT solutions. The counter-argument provided in a recent Gartner study[9] citing architecture, processes and practices of the best high-tech firms as the ways worth emulating is rather incomplete. To better answer this criticism we can again point to the level of abstraction. The architecture of uniform resources is scale-invariant and therefore accommodates businesses large and small alike.

Enterprise Software Systems below and at Web-scale

Today most enterprise software systems do not truly achieve Web-scale nor Web-orientation as described above. The software solutions, not the Web solutions, used by Facebook, Google and Amazon are no different in this respect.

Software that in fact does achieve Web-scale is based on *Resource-Oriented Computing*. *ROC* was developed to avoid the economic saw-tooth pattern in enterprise software systems and to provide Web-scale properties. It literally applies the Web-scale abstraction to software itself by treating everything from code to information to computational results as uniform resource. Two visualizations help compare (a) source code as most of it has been written for the last 40 years with (b) a Web-scale system of uniform resources called

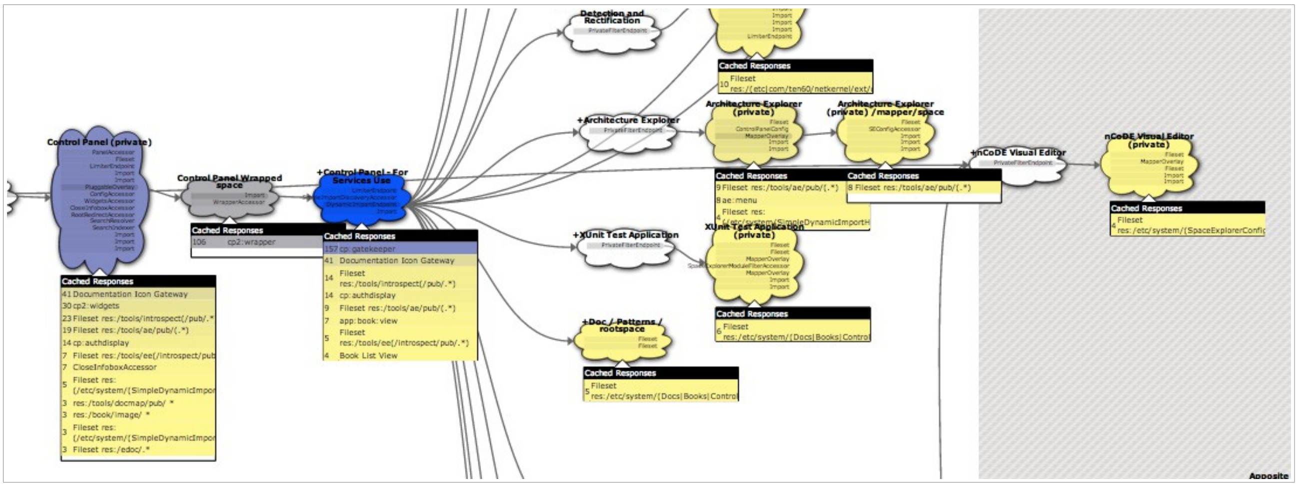
8 MIT Sloan http://sloanm.it/182Itca
9 Information Management, Gartner: Global Enterprise Increasingly Adopt Web-scale IT. http://bit.ly/1h9qjum



NetKernel.

(a) Below Web-scale, the traditional computer program: columns of source code comprising tens or hundreds of thousands or several million lines of code.[10] The way code itself is organized in this system does not approach the efficiency with which the Web structures information. Note also, **code and information are all mixed together.**

(b) At Web-scale, the NetKernel uniform resource engine: Screen shot from the "Cache Heatmap" shows a live system made up of uniform resources. Developers can apply changes to the system in this window in real-time. Colors indicate live performance characteristics of the entire application system. In case of a crash or error in the system, it captures system state to allow for fast error analysis and efficient repairs. This *uniform resource engine* load balances itself across servers and multicore. **Code and information are kept separate at all times.**



Computing Footprint & Cloud

Power and cooling costs are linked to the price of coal and rise faster than inflation.[11] That makes it all the more important for data center customers to use every technology at their disposal to reduce power consumption in the Cloud by minimizing their computing footprint. The one additional lever that Web-scale Resource-Oriented Computing can offer

10 University of Groningen Computer Science Department www.cs.rug.nl/svcg

11 Coal 4-Year Low Lures Utilities Ignoring Climate: Energy Markets. Bloomberg.com 11 Oct 2013.

here, in contrast to object-or service oriented systems, is that it minimizes power consumption by design. Its built-in caching mechanism is used to optimize the rate of change in state of all uniform resources combined.

Web-scale means Quality means Market Share

If Web-scale IT is the metaphor by which we can describe the means, then enterprise software systems of the highest quality are the goal. Building Web-scale IT systems means putting a software architecture in place that IT departments can rely on and trust. Software systems built at Web-scale provide a higher level of quality and structural integrity.

Quality [12]	Traditional	ROC
Reliability		
Monitoring entire *live* system state	No	Yes
Visualizing *live* system state	No	Yes
Capturing entire system state at point of failure	No	Yes
Integrated testing tools	n/a	Yes
Efficiency		
Reuse of code / resources	Low	High
Micro-caching (reduces computing footprint)	No	Yes
Scalability over multi-core	Declining	Linear
Self-load-balancing code base	No	Yes
Architecture paradigm	Asymmetrical	Symmetrical
Security		
Contextual resource access	No	Yes
System audits at resource level	No	Yes
Physical separation of code, information & state	No	Yes
Maintainability		
Visual development tools	Yes	Yes
Object Relational Impedance Mismatch	Yes	No
Separation of Dev / Ops	Yes	No
Complexity / complication	Scale variant	Scale in-variant
Orthogonality of components	No	Yes
Software virtualization	No	Yes
Data binding (reduces long-term maintainability)	Yes	No
Coupling of components and computations	Loosely coupled	Decoupled
Middleware in distributed systems	Yes	No
Adding System Constraints to Architecture	Before	After
Uniform components	No	Yes
Size		
Component Average Number of LOC	200-400	< 200
Size of download	n/a	30MB

12 CISQ Quality Model defines 5 characteristics in software systems: reliability, efficiency, security, maintainability and size. One-for-one comparisons between fundamentally different approaches to computing are impossible. We therefore list characteristics as best proxies in lieu of direct comparisons.

The Competitive Advantage of Web-scale

Resource-oriented enterprise software systems at Web-scale provide IT departments with these capabilities:

Aligning IT to Business	High rate of resource re-use
Protecting Investments	Extended useful life of software
Improving Systems Quality	Greater structural integrity
Lowering Power Consumption / Cloud	Reduced computing footprint
Reducing Total Cost of Ownership	Economies of scale inside Web-scale software systems, life system information

In summary, resource-oriented and Web-scale software systems allow CEOs/CIOs to make short- and long-term investment decisions with confidence, create value for their businesses and protect it in the long term.

To find out how your company can extend the useful life of its enterprise software systems or to simply obtain more information send an email to **sales@1060research.com** or call **+1 862-772-1060**

About the Company

1060 Research Ltd. was spun-out from HP Labs in Bristol in 2002, pioneered a new abstraction in computing called *Resource-Oriented Computing*® in order to change the economics of software engineering. Its team of architects developed the *Uniform Resource Engine*™ called *NetKernel*® which has been deployed in some of the most demanding environments and in a number of industries including telecommunications, e-commerce, defense and education.