



## Creating the cumulus

Software will be transformed into a combination of services

**W**ANT to become a programmer in 20 minutes? With the program from Iceberg, a start-up, you can. Just open the "getting started wizard" and pick the type of application, say "project management". Type in its "business objects"-things like "client", "team" and "members"-and tell the tool how they relate to each other. Then design some input forms and define the process for getting a project done. Another click and you are ready to go.

In reality, of course, things are not that simple. And many professional programmers will scoff at the development tool as a mere toy. Yet Iceberg and similar outfits demonstrate that geeks are losing their monopoly on programming. Now, with a bit of patience, anybody can create a simple application, for instance, to collaborate with colleagues or to draw an online map.

This democratisation of programming, however, is only a small part of something much deeper: a fundamental change in the nature of software. It is not just that more and more software will become a service delivered online. More importantly, applications, web-based or not, will no longer come as a big chunk of software, but will be made up of a combination of electronic services—a shift that has picked up a lot of speed since computing began moving into the cloud.

To understand this new way of building applications, known as "service-oriented architecture" (SOA), think of a culinary analogy. Whereas the old chunk of software resembles a precooked meal that just has to be popped into the oven, the new architecture is more like a restaurant. It is a service in itself but also a combination of sub-services. There is the waiter

who takes the order and conveys it to the kitchen. There is the cook who prepares the food. And there are the cleaners who keep the place tidy. Together they create the "application": a restaurant.

### An attack of the vapours

The importance of this shift from a monolithic product to services is hard to overstate. In a sense, it has seeded the cloud, allowing the droplets—the services that make up the electronic vapour—to form. It will allow computing to expand in all directions and serve ever more users. The new architecture also helps the less technically minded to shape their own clouds, using such tools as Iceberg's.

Just as for the industrialisation of data centres, there is a historic precedent for this shift in architecture: the invention of movable type in the 15th century. At the time, printing itself was not a new idea. But it was Gutenberg and his collaborators who thought up the technologies needed to make printing available on a mass scale, creating letters made of metal that could be quickly assembled and re-used.

Similarly, the concept of modularity has been around since the early days of computing. "Everything in computer science is to just write less code. What is the technique for writing less code? It's called subroutines," said Bill Gates, Microsoft's founder, in a recent interview. A subroutine is a part of a program that can be re-used, just like movable type. The idea, says Mr Gates, has always been to apply this principle of a subroutine more broadly.

Yet this did not happen, mainly because the cost of computing fell much faster than that of communications. Ever

cheaper and more powerful chips made it possible to move from mainframes to minicomputers to personal computers (PCs) and now to hand-held devices. But connecting all these pieces remained difficult and expensive, which meant that such devices all had to come with their own data and chunky programs. Now, thanks to plenty of cheap bandwidth and more and more wireless connectivity, computing is able to regroup into specialised services, or Mr Gates's subroutines: "We now live in a world where...[a] subroutine can exist on another computer across the internet."

Part of Gutenberg's genius was to recognise the need for all the letters to be identical in height so they could be easily combined. Similarly, for computing services to work there had to be robust technical standards. Only a few years ago this seemed far beyond the IT industry's reach. Most firms insisted on their proprietary technology, mostly to lock in their customers. Again, cheaper communications helped to bring about change. The success of the internet demonstrated the huge benefits of open standards and forced vendors to agree on common ways for their wares to work together. One result is a stack of something called "web-services" standards.

Service-oriented architecture first showed up in open-source software but was quickly adopted by big enterprise-software vendors because they had a pressing need for it, says Jim Shepherd of AMR Research, a consultancy. Big software vendors, for instance, had to find a way to untangle the hairball of code that their products had become, or else they themselves would choke on it. Customers wanted more flexible and extensible programs.

Think back to the gastronomic example. A precooked meal is hard to change, and so are traditional software applications. By contrast, a restaurant can easily change its menu and its style of operation. Similarly, soA-based software allows companies to alter their business processes, such as the way they handle orders to collect cash.

SAP, a German software firm, was one of the first companies to put this service-oriented architecture front and centre. Starting in 2003, it developed, among other things, a new corporate-software package that did away with monolithic applications, such as programs to keep track of a company's finances or manage its relationship with customers. Instead, it introduced a collection of re-usable components that could be strung together at will.

IBM, too, is a fan of SOA and web services. But its approach is somewhat different, given that it does not sell business applications but makes most of its money from IT services and software to manage the underlying computing infrastructure. IBM uses SOA mainly to help firms integrate their increasingly complex and disparate IT systems. Its software turns them into a collection of services that can be woven into business processes.

The approaches may be different but the vision is the same: to create IT systems that adapt to the business needs of companies and allow them to connect. "When I want to do something new", explains Steve Mills, the boss of IBM's software group, "I do not need to build a new application but can use the pieces I already have." To Peter Zencke, who led the development of SAP's new package, its most exciting feature is that "any of the process components can now become a service provided by some other firm."

Despite millions of dollars spent on marketing SOA, it has not really taken off yet. But many web-based applications for consumers rely on this concept. The prime example is Google Maps. When the online giant launched the service, programmers quickly figured out how to mix the maps with other sources of information. This is how, for example, Housingmaps.com was created, a combination of a Google map with the rental and sales listings from Craigslist, a website for classified ads. It was one of the first "mash-ups", as such combinations have come to be called.

Since then the number of such mash-ups has exploded, thanks mainly to services like Microsoft's Popfly and Yahoo! Pipes. In essence these are graphical pro-

gramming tools. Users drag and drop "modules"-data feeds providing such information as pictures, headlines and search results-and weave them together.

Most of these mash-ups are still toys, but firms offering software as a service have started offering similar combinations. In April Salesforce.com and Google announced that they would integrate their online services. Users of Salesforce, which helps firms manage their customer relationships, can now quickly switch to Google's web-based applications.

Smaller firms have already started to weave a network of services. OpSource, a Silicon Valley start-up, for instance, provides basic services for other saas firms and web companies. TriCipher, another Californian newcomer, authenticates users of web applications. Ribbit, for its part, allows these services to add voice communications to their offerings.

Yet it is unlikely that the software cloud will end up as a vast nebula of thousands of specialised services. Even creating a service-oriented architecture is "no silver bullet" against complexity, in a famous phrase by Frederick Brooks, an elder of computer science. Although web services allow online offerings to connect, for instance, it is costly to synchronise their data. And it would not make sense for any firm to bet its business on simple mash-ups.

As software markets mature, they tend to form two kinds of clumps: integrated suites of applications, and platforms on top of which others can build programs. Both forms are already emerging. On the applications side there is Google Apps and

Zoho, which is even more comprehensive. It encompasses a total of 18 applications, including word processing, project management and customer-relationship management (CRM).

As for platforms, there are already plenty, in different shapes and sizes. For enterprise applications, SAP has built one called Netweaver. Oracle offers something similar called Fusion. Last year, Salesforce launched a "platform as a service", allowing other firms to use the plumbing that supports its own CRM offering.

More recently platforms for consumer services have been proliferating. Facebook, a social network, was the first to become one in 2007. Other big online firms have followed suit or will do so soon: Google with App Engine, Yahoo! with Y!OS and Microsoft with a "cloud operating system" thought to be called Windows Strata. Some predict a platform war to rival the epic fights between Microsoft's Windows and Apple's Macintosh.

### Never say die

What shape will the software cloud take, other than being a vast collection of services? In one way it will look much like the old software world. There will be a few big platforms, akin to today's operating systems, and most applications will be written to one of these platforms.

What is less clear is just how much of business and consumer software will migrate into the cloud, and how fast. The answer depends on whom you ask. Unsurprisingly, Marc Benioff, Salesforce's founder and chief executive, argues that web applications will spell the "death of software". But people are not about to throw out their powerful PCS or other "client" devices, if only because many of them still work offline at times. Similarly, companies will always want to keep some applications in-house, for reasons of security, regulation or simply to maintain control. Ray Ozzie, Microsoft's chief software architect, promotes something called "software plus services", meaning that customers will settle on "the right mix of old and new stuff".

If history is any guide, Mr Ozzie is more on the mark. Even the biggest changes in IT have never spelt the death of anything, notes Josh Greenbaum of Enterprise Applications Consulting. IBM, for instance, is still making money with mainframes.

So the software cloud, just like its hardware underpinnings, will be very diverse. But how will people make use of this kind of computing? •

