PROCEEDINGS OF SCIENCE

Bioscience in the Cloud - Next Generation Analysis for Next Generation Sequencing

Jens Jensen[1]
STFC
Harwell Oxford Campus, Oxon OX11 0QX, UK
E-mail: `jens.jensen@stfc.ac.uk`

Nick Trigg
Constellation Technologies
STFC Innovations, Harwell Oxford Campus, Oxon OX11 0QX
E-mail: `nick.trigg@constellationtechnologies.com`

Jonathan Churchill
STFC
Harwell Oxford Campus, Oxon OX11 0QX, UK
E-mail: `jonathan.churchill@stfc.ac.uk`

The business drivers facing the life science industry that are behind this body of work are true for all companies in the sector; large and small. They are increasing costs in IT, increasing data volumes, increasing numbers of models/applications/tools being produced by research and the urgency to push results from early stage research to revenue. Although these pressures are true throughout the field of bioinformatics, the particular concern is in the relatively new area of Next Generation Sequencing; it shows great promise for science but comes with significant IT challenges.

This paper describes technical work on setting up and securing PlasMapper and Ensembl with federated access management. The work further combines these services with resources in Microsoft Azure, using WS-Federation as authentication. The project was extremely challenging in aiming for high targets and quick delivery on a tiny budget, so had to reuse existing components. We found that some of these were considerably less mature than expected, and, perhaps unsurprisingly, adherence to standards was in some cases rather lax.

We achieved a lot given limited funding, and anyone aiming to integrate public and private clouds, or securing services for genomics, or building WS-Federation services, will be interested in this work.

The International Symposium on Grids and Clouds and the Open Grid Forum,
Academia Sinica, Taipei, Taiwan
March 19 - 25, 2011

[1] Speaker

1.Background

The work described in this paper was funded by the Pistoia Alliance [1]:

> "The Pistoia Alliance is a global, not-for-profit, precompetitive alliance of life science companies, vendors, publishers, and academic groups that aims to lower barriers to innovation by improving the interoperability of R&D business processes. We differ from standards groups because we bring together the key constituents to identify the root causes that lead to R&D inefficiencies and develop best practices and technology pilots to overcome common obstacles. Pistoia-led projects have the potential to transform precompetitive R&D, making it more open and collaborative and providing a platform upon which organizations can innovate in the ways most suited to their ultimate goals."

Realising that the ability to generate vast amounts of data throughout the R&D process of drug discovery and clinical trials is going to be a massive challenge [2], the members companies are trying to develop cloud and security standards for the sector.

This particular project specified the delivery of a secure cloud system to enable companies to compare their own gene sequences with those in Ensembl [3] and to search across their own published sequences. Also PlasMapper [4] was deployed with the same level of assurance. The system had to be secure to the highest standards and was going to be test hacked. Additional functional features were also required. The whole system was live tested by several different life science customers under real use conditions.

The partners within the system development consortium were Constellation Technologies (a bioinformatics company specialising in large data sets and heavy compute services in the life science industry - "bioinformatics on the cloud"), STFC (one of the UK's leading national labs specialising in grid and cloud computing and cloud security), Active Web Solutions (a UK based Microsoft partner specialising in solutions in the Azure cloud), and Microsoft Azure.

The unique features of Constellation's solution were

1. Demonstrating end-to-end security in access to Ensembl and additional tools
2. The user could use either Linux or Windows applications as required
3. Easy to use
4. Secure API into Ensembl
5. Allowed academics free access to the service

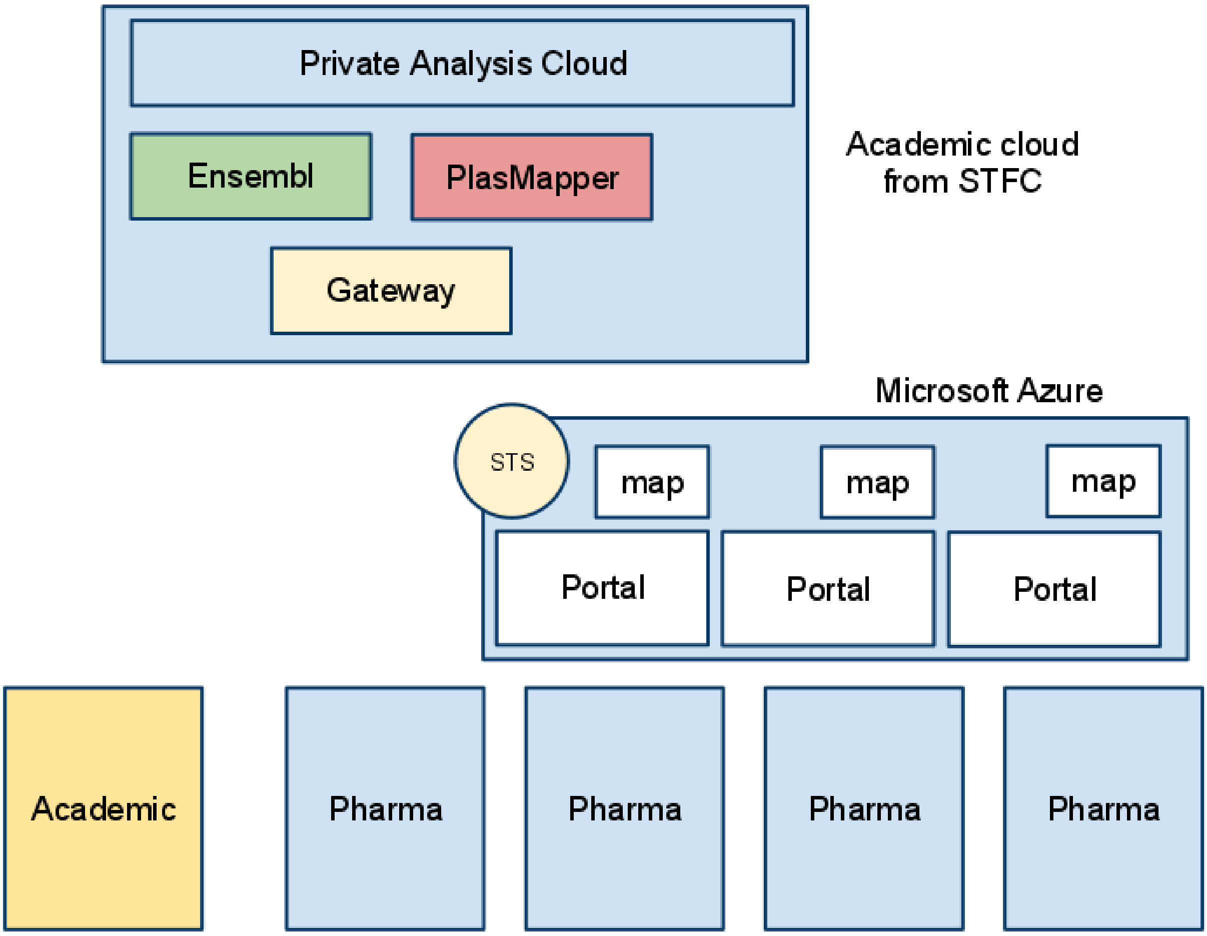
The outcome is noteworthy also for seamlessly unifying commercial clouds (Azure) with academic resources. This will be extremely useful in future work because the commercial and academic clouds have different capabilities: they are resourced, certified, and priced differently. By combining the two, we open up possibilities for getting the best of both worlds, to the ultimate benefit of the end users. Further details of the work can be obtained from [13].

1.1 Related work

During the lifetime of this project, Microsoft announced the availability of an open-source BLAST implementation for Azure ([5],[6]). Moreover, we also know of a front-end to this based on Microsoft Excel [7].

2.Architecture

The following diagram sketches the architecture of the system:



The general idea behind the architecture is to provide services managing confidential data within Azure, relying also on the certification of the Azure data centre to help reassure the customers from the life sciences companies that their data is adequately protected. The academic cloud provided by STFC manages the publicly available resources, but with added security.

The primary set of users are from the life sciences companies: each company has its own portal running inside Azure, with a specialised database containing annotations. Each portal is separate from the others, so a user from one company would have no means of seeing what a user from another company is doing. Also in Azure we find the Secure Token Service, or STS. A component of a WS-Federation architecture [8], the STS manages users' credentials.

Further up the diagram we find the Ensembl and PlasMapper resources provided by STFC - also provided here but not shown are certain databases containing publicly available annotation data. Using MySQL, these databases are accessed directly by the Ensembl API (described below.)

While the purpose of this project is mainly to demonstrate end-to-end security, it is now possible for STFC to scale the resources further by adding high performance and high throughput computing resources. This would enable us to lift restrictions in Ensembl and PlasMapper on the size of data being processed and the number of concurrent users.

For expert users, Ensembl provide tools for making improved use of Ensembl. Known as the "Ensembl API," it is actually a set of Perl modules which can be called from the user's own Perl programs. We provided such modules for users at the life sciences companies to access the Ensembl database directly in a secure way, without requiring a portal login.

3.Implementation

The focus in this paper is mainly on the open source parts of the infrastructure: the components and the "glue," and the result.

The first observation was that PlasMapper and Ensembl were not built to be secure services: indeed, the portals are usually used online over HTTP. We thus decided:

1. To make few changes - ideally none - to Ensembl and PlasMapper themselves, to minimise the amount of work required in upgrading;
2. As both Ensembl and PlasMapper are Open Source, to stick with OS code as far as possible, so we could contribute the work back to the communities;
3. Private network from the gateway to Ensembl and PlasMapper and from Ensembl/PlasMapper to the private analysis cloud built on top of Platform LSF;

It follows that the Ensembl and PlasMapper services were best secured with a secure gateway: a front end which manages the security, and passes all requests onto the backend services over secure networks. Data from PlasMapper and Ensembl were modified to ensure that data is present in the private cloud only for the duration of the analysis.

The gateway security would have to be set up to allow secure access precisely from three sources:

1. The life sciences users, via the Azure portal;
2. The life sciences users directly, using their Ensembl APIs;
3. Academic users.

"Secure access" means primarily authenticated and with confidential access. Authorisation and accounting will usually be done by company. In fact, by the time the user's requests reach the academic services, they will have been anonymised to the extent that we know which company they came from, but not which employee.

To ensure a consistent security infrastructure, we would ideally link authentication into the gateway with the Azure STS: an attempt to access the service would direct the unauthenticated user back to the STS using WS-Federation in passive mode, similarly to Shibboleth redirects. However, users of the API would then need to authenticate by the same means, obtaining local credentials from the STS first, or as they make their first access from within their Perl programs - or alternatively, they would have to authenticate to the gateway by some other means.

3.1 Rewriting

As the gateway, we used an Apache web server, in reverse proxy mode, meaning that requests to the backend services would have to be relayed to the appropriate service based on the URL path. In this respect, PlasMapper was relatively straightforward: already hosted under /PlasMapper by a Tomcat hosting environment, the gateway was configured to relay anything with a path starting with /PlasMapper to the Tomcat service, rewriting only the hostname. Ensembl was not so easy. It uses numerous paths inconsistently, including one per species. We tried different approaches:

1. Using a specific path prefix "/ens/", so a user request for "/ens/foo" would appear as a request for "/foo" to Ensembl;
2. Adding every path prefix, including one for each species, one for images, one for scripts, etc., to the gateway's redirect tables;
3. Redirecting everything that is not for PlasMapper to Ensembl.

Using mod_proxy_http to redirect URLs is not sufficient: the returned HTML must also be rewritten using mod_proxy_html or the clients will get broken URLs. As Ensembl relies not just on traditional HTML but also on JavaScript to render its pages, rewriting rules have to be crafted carefully. For example, an early attempt to rewrite an initial "/" path within scripts to "/ens/" (in path strategy 1 above), script comments "//" got mechanically rewritten to "/ens//ens/", thus breaking all the scripts.

While the production service we deployed was based on rewriting strategy 1, and we had all three working (in the sense of having "most" of the Ensembl service working), it would probably have been better, or at least quicker, to have used strategy 3. One might think this strategy can have security issues since there is no filtering of the URLs being passed from an authenticated client to the backend service, but this is probably not a great concern given that URLs are passed along based on their path prefix only, and that clients have to authenticate anyway.

3.2 Authentication

As the gateway manages authentication on behalf of the services behind it, integration with the cloud makes it necessary for it to appear as a WS-Federation resource. Staying with Open Source code, we picked the PingIdentity WS-Federation module for Apache. Unfortunately, the code is not current: last change was in 2005, and WS-Federation now uses different XML namespaces [9], and slightly different attributes appear in the SAML sent by the STS. Perhaps worse, the code had some bugs, including not escaping redirection URLs properly, and sending incorrect timestamps. While these were fixable, they took a while to fix as STS would not always tell us what the problem was, so due to time constraints we abandoned WS-Federation for the gateway and made both the portals and the Ensembl API use X.509 client certificates. Nevertheless, it would have been perfectly possible to persevere and get WS-Federation fully working.

4.Security Considerations

Within this project, users were authenticated by institution: it did not matter much if people within each life sciences company shared credentials, as the aim was to protect one company from the other, not users from each other. (The implication of revoking a certificate is then obviously to revoke access for everybody in the whole company, and we would lose individual traceability, but these were considered acceptable risks for this project.)

For the API (i.e., Perl modules), we packaged them up separately to contain certificates. Likewise, certificates could easily be deployed within each portal in Azure. The client certificates were created from one of our own CAs - there is in general no need for the client to trust this CA, as they only use it to access the server. The server certificates were obtained from a commercial CA, to ensure that they were trusted also if users use browsers to access the services.

The client credentials (i.e., the private key) were encrypted using standard (password-based) encryption. Moreover, the APIs were distributed to the users using randomly generated URLs: in this way, we could mail them the URLs which they use to download the Perl code, and then text them the password for the private key, or communicate by some other out-of-band means.

As part of the Pistoia Alliance call, the infrastructure had to be evaluated by an “ethical hacker,” with a remit to test everything except denial of service attacks (to which the current infrastructure is quite vulnerable once a user is authenticated - the project aims to demonstrate confidentiality but not high availability of services.) As the gateway requires certificates, part of this test also involved giving a certificate to the “hacker” to see if they could then discover something they shouldn’t - accessing the service without the certificate was not possible.

The main security concern for this part of the project was usability, rather than confidentiality. It is easy to secure something extremely well by requiring specific client certificates, but users will need to manage these, and the user tools for managing certificates often give rise to usability issues. A secondary concern were the functionality of the service: it turns out that there are parts of Ensembl which expose certain vulnerabilities: e.g., it does not validate whether its input is malicious code (which is quite natural, as it was not built for security.) Some of these we were able to switch off in our implementation, but the implication for the legitimate user is then loss of functionality.

A final “interesting” experience was a security vulnerability [10] in secure sockets which had made OS vendors turn off renegotiation (pre-RFC5746 [11]). As the service relied on renegotiation, we had to replace the system libraries, choosing to replicate the system with RFC5746 compatible libraries in a chroot’ed directory. However, unpatched browsers then refused to work, returning generally misleading error messages (such as “you are not connected to the Internet.”) We discovered quite a few unpatched browser installations, even in early 2011.

5.Next steps

This project proved that it was possible to build a low level bioinformatics service on a secure cloud that could be used by the real users, linking public clouds (Azure) and private (STFC’s).

The next steps will be to add additional services which will build the service into a real system that industrial users will be interest to use as part of their normal course of business. This is certainly possible notwithstanding the industry’s concerns about security. A full WS-Federation based authentication/authorisation throughout public networks, integrated with Active Directory Federation Services in each life sciences company would also be an interesting step, so even expert users may no longer need certificates.

An additional next step is to develop a wider set of services, possibly investigating interoperation based on existing standards and data formats [12], making them available to “on-demand” users and bring to the end-users the real power of the cloud - new computer models, increased infrastructure and on-demand pricing. This is the ultimate goal of bioinformatics on the cloud.

Acknowledgments

The project is grateful to the Pistoia Alliance for funding the project, and for their valuable comments and feedback on this work. The full project is a collaboration between Microsoft Azure and Active Web Services, and the authors of this paper acknowledge their work in the full project – for reasons of space, we have been able to cover only a part of the work in this paper, but hopefully sufficient to give the reader an idea also of the full picture. The reader interested in further information should contact Constellation Technologies [13].

6.Conclusion

The life science sector is facing some significant IT challenges going forward. New research techniques are generating much larger amounts of data and new research initiatives are developing new and sophisticated computer modelling techniques. This progress is very exciting from the biology point of view and promising some exciting medical advances across the whole sector. However, one of the critical path issues between now and this exiting future is an IT problem rather than a biology problem and for industry players to take advantage of these breakthroughs they will need to use computers in new ways.

This project showed that it is technically feasible to provide a services that allows users to access software on the cloud allowing their personnel access to a huge range of analysis tools/applications. This area is particularly exciting to large life sciences and smaller biotech as it allows companies of all sizes to use this “wall” of data heading their way. The world of biology only gives up its secrets slowly as it is particularly complicated. Large scale computing will not solve all of the problems but will enable the life science researchers to spend more time thinking of the biology rather than the IT element of the process.

This project integrated private cloud and commercial (public) cloud resources to provide services for genomics for the life sciences and pharmaceutical industry. Using open source tools such as Ensembl and PlasMapper, and open data, a specific set of genomes, we built a security framework around services to ensure the confidentiality of the analysis performed by users from life sciences companies. Each of the companies were fully segregated within the Azure cloud, and using the certification of the Microsoft datacentre we were able to convince them to place some sensitive data in the cloud. Access was both for users using portals and for people using the so-called Ensembl API, a secured version of which was also deployed. Finally, we had access also for academic researchers, although in this limited scale demonstrated it only for researchers within STFC.

The promise of the project is a future where resources can be obtained on demand, where we move towards trusted clouds that can cope with the challenges of next generation sequencing, as well as other sciences with high data volumes.

References

[1] Pistoia Alliance – www.pistoiaalliance.org

[2] S. C. Schuster: *Next-generation sequencing transforms today's biology*, Nature Methods - 5, 16 - 18 (2008) , DOI:10.1038/nmeth1156

[3] T Hubbard, *et al.*: *The Ensembl genome database project*, Nucl. Acids Res. (2002) 30 (1): 38-41. DOI: 10.1093/nar/30.1.38

[4] B. Stone, G.L.Griesinger, J. L. Modelevsky: *PLASMAP: an interactive computational tool for storage, retrieval and device-independent graphic display of conventional restriction maps*, Nucl. Acids Res. (1984) 12 (1Part2): 465-471; DOI: 10.1093/nar/12.1Part2.465

[5] R. Barga: pers. comm. (UK e-Science All Hands, Cardiff, 13-16 Sep.2010)

[6] W. Lu, J. Jackson, R. Barga: AzureBlast: *A Case Study of Developing Science Applications on the Cloud*, Proceedings of the 1st Workshop on Scientific Cloud Computing (Science Cloud 2010)

[7] Microsoft Research Sequence Assembler: in http://research.microsoft.com/en-us/projects/bio/mbf-sample-apps.aspx (retr. Apr.2011)

[8] M. Goodner, M. Hondo, A. Nadalin, M. McIntosh, D. Schmidt: *Understanding WS-Federation*, (2007), available from e.g. http://msdn.microsoft.com/en-us/library/bb498017.aspx (retr. Apr.2011)

[9] WS-Federation 1.2 Specification: http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.pdf (retr. Apr.2011)

[10] http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-3555 (retr. Apr.2011)

[11] E. Rescorla, M. Ray, S. Dispensa, N. Oskov: *Transport Layer Security (TLS) Renegotiation Indication Extension*, RFC 5746, www.rfc-editor.org/rfc/rfc5746.txt

[12] C. F. Taylor: *Standards for reporting bioscience data: a forward look*, Drug Discovery Today, Vol.12, issue 13-14 (2007), pp.527-533

[13] N. Trigg, *et al.*: *Secure Gene Sequencing*, Constellation Technologies, http://www.constellationtechnologies.com/