

## TECHNOLOGY CHOICES FACTSHEET

As a consequence of our discovery work, including extensive consultation and research with experts, the legislative drafting, amending and publishing tools project has made some provisional technology choices. In essence we are looking to build a drafting tool that is predominantly based on leading open source technologies and open standards – the Akoma Ntoso XML markup language, the eXist XML database, the MEAN stack of JavaScript frameworks (Mongo, Express, Angular, Node) and a customised XML/HTML editor. We believe this mixture of technologies provides the best chances of success, to tackle the unique challenges of drafting and amending legislation documents.

This factsheet gives a little more information about the technology choices we have made.

### Markup: Akoma Ntoso XML

- We aim to use XML as our base markup language. It is tried and tested and ideally suited to the structured nature of legislation, and can easily be transformed using open technologies into other formats. It is also likely to be in use over the longer term.
- Specifically we aim to use Akoma Ntoso rather than the Crown Legislation Markup Language (CLML). Akoma Ntoso is the emerging international standard for representing legislation, and is less complex and easier to work with than CLML. There is also a wider pool of experts and suppliers.
- We have recently extended the legislation.gov.uk API to support Akoma Ntoso. Any piece of legislation can be accessed in this format, by adding /data.akn to the URL. For example: Budget (Scotland) Act 2015 can be viewed in this format at: <http://www.legislation.gov.uk/asp/2015/2/enacted/data.akn>
- We have also serialised this Akoma Ntoso XML as HTML5 for online presentation. This can be accessed by adding /data.html to the URL. For example: <http://www.legislation.gov.uk/asp/2015/2/enacted/data.html>

### Backend: eXist XML database and Mongo JSON database

- We aim to use an xml document database rather than a relational or other kind of document database because it is best suited to holding structured, document-based data such as Bills and amendments. An XML database can hold XML or HTML (as XHTML) natively, and interact with it in highly complex ways using open standards such as Xquery.
- Our intention is to build a system with a clear separation between the server side (i.e. the backend) and client-side (i.e. the front-end) with all interaction taking place via a well-defined API layer.
- The eXist database is the leading open-source XML database and is ideally suited to a production environment.
- However, an XML database is unlikely to be best suited to holding all the types of data required. We believe that to deliver the kind of user experience we are aiming at, we need to use a server-side database that can be closely synchronised with data held in the browser, to store data about the structure of the document and cross-references. Our investigations indicate that the open-source MongoDB database, or

something similar, would be effective, given its close integration with the JavaScript frameworks we are considering.

### **Middleware: Node JavaScript Framework**

- Node JS is an open-source JavaScript platform designed for building fast, scalable network applications, particularly data-intensive real-time applications that run across distributed devices. It is widely supported and forms part of the MEAN stack. However further investigation will be required to confirm that Node works well with XML and the eXist database.
- An alternative is to use the open-source Orbeon Forms framework – however Node's focus on web applications and links with other JavaScript frameworks makes it our preferred choice at this stage.

### **Front-end/client: angular JavaScript Framework and customised XML/HTML editor**

- AngularJS is an open-source JavaScript platform for creating dynamic web pages and applications. Given that it is an industry-standard technology with proven use in similar sectors, we think using Angular, as part of the MEAN stack, is the appropriate way forward.
- In addition we will require a highly customised XML or HTML editing component that is applicable to the task of authoring or editing legislation.