MOSAIC

Science, Humanities, and Engineering Portal

MOSAIC is a cloud portal for hosting scientific, academic, and technical publications. MOSAIC bridges the gap between traditional academic publishing and the new world of software- and data-driven research platforms, which feature interactive reading experiences and open-access data sharing.

Today there are many Academic/Scientific/Technical Portals. These resources, catering to an academic and technical audience, aggregate research articles, books, journals, and publicly accessible research data. AST Portals typically link to web pages devoted to individual publications, with each page displaying citations, abstracts, keywords/topical classification, and (when appropriate) links to download documents in e-reader (typically PDF) formats. Some AST Portals are curated by individual publishers; others (such as SciVerse, Mendeley Data, and Dryad) are scientific projects incorporating content from many publishers.

0.1 *The MOSAIC Architecture and Data Models*

A MOSAIC portal combines a central web service with a collection of independent Research Objects. Not every document indexed by MOSAIC must have an associated Research Object, but the technological design of MOSAIC prioritizes integrating research portals with downloadable Research Objects.

In particular, MOSAIC encourages the creation of *self-contained* Research Objects. MOSAIC assumes that the typical Research Object indexed on a MOSAIC portal will be a downloadable code-and-data bundle with *minimal external dependencies*. It should be easy for readers to get started exploring and interacting with Research Objects without complicated downloads or installation of additional software. Authors are certainly free to *enhance* Research Objects with content tailored to specialized software — for example, sophisticated data visualization features, or integration with software commonly used in the relevant research fields. However, these extra capabilities augment the core of the RO, which should provide a standalone mechanism for researchers to understand, visualize, and reuse the bundled code and data, without requiring external software or code libraries.

In order to help authors create self-contained Research Objects, MOSAIC defines a *Reference Application Kernel* (RAK) which presents a canonical format for bundling research data and code into a self-contained package. Each RO bundle using the RAK format provides a standalone, desktop-style Dataset Application that offers an entry point to the raw data. By design, it should be easy to download the RAK bundle from MOSAIC and then compile and launch the Dataset Application, so readers can begin exploring the data set with minimal hassle.

The RAK format includes other features as well, such as code export and testing, which provide a more detailed access to the published data (including several command-line executables built alongside the Dataset Application). However, the Dataset Application presents a useful introduction to the underlying data set, which helps users get oriented to the data set in its broad overview before they start to examine it in finer detail.

Technically, MOSAIC RAK bundles are code repositories based on the Qt platform for C++ Applications. In RAK, each data set features C++ code which has no dependencies apart from Qt itself. The Dataset Application should compile and run automatically from inside Qt Creator. Because Qt is an advanced desktop GUI development platform, it allows ROs to present compelling, interactive graphics and GUI components — including 2D and 3D visualizations, tables and charts, context menus, dialog boxes, explanations of technical terms, embedded PDF viewers, and other features which make Dataset Applications an interactive and pedagogically useful tool for exploring raw data, customized for each data set.

Research Objects indexed by MOSAIC do not have to use the RAK model. However, RAK provides a Reference Implementation demonstrating how RO metadata can be described for it to be automatically processed by a MOSAIC portal. Developers using different RO models can examine how RAK bundles expose data to MOSAIC so as to create analogous metadata for their own bundles.

0.2 *The MOSAIC SDK and Data Modeling Features*

To facilitate implementation of RAK bundles, MOSAIC provides several development tools and code libraries which can help set up a development environment tailored to the MOSAIC ecosystem. These assets include the following:

Build Tools Since it is designed to run inside Qt Creator, the RAK build system is based on *qmake*, which is Qt's layer atop the C/C++ "make" system. MOSAIC's build system extends *qmake* to simplify the integration of RAK applications with MOSAIC portals and with other scientific software that may be available as enhancements for RAK applications, depending on discipline/topics.

Hypergraph Code Libraries and Parsers MOSAIC defines a multiparadigm representation for scientific data based on Directed Hypergraphs. This representation is flexible enough to accommodate many different modeling paradigms — including conventional OWL Ontologies, CSML-style Conceptual Spaces, and several other workflow and linguistics-based models

— while at the same time conducive to self-contained Dataset Applications. The MOSAIC SDK provides code for creating hypergraphs from text documents and integrating hypergraph data into C++ applications.

Scripting and Testing The MOSAIC SDK includes components that developers can use to create test suites for Research Objects and to design a customized scripting environment. Selected functions from the RO code are exposed to a Runtime Reflection engine, which allows these functions to be invoked by scripts, including scripts composed for unit and integration testing. Functional annotations provide detailed Requirements Engineering for application procedures, including scientific details like statistical scale (Nominal/Ordinal/Interval/Ratio), dimensions, ranges, and declarations of side effects.

MOSAIC Interop Applications built with the MOSAIC SDK can also automatically interoperate with MOSAIC portals. The SDK allows Dataset Applications to be equipped with special features for different user roles, including authors and editors as well as ordinary readers. Authorized users can automatically notify MOSAIC, from within their Dataset Application itself, about a new or updated version of their Research Object.

Architecturally, MOSAIC is developed as a Cloud/Native Hybrid featuring a collection of independent native applications (the RAK Research Objects) connected to a central cloud service (the MOSAIC portal). The portal itself is also a Cloud/Native Hybrid, in that a miniature version of a MOSAIC portal can run as a standalone desktop application. MOSAIC is implemented to allow seamless integration between native and cloud components — for example, using non-GUI Qt libraries as the core of the server-side code — so as to minimize data marshaling when the cloud service communicates with native endpoints.

As described above, not every Research Object indexed by a MOSAIC portal will use the MOSAIC SDK or MOSAIC's RAK format. Research Object developers are free to adopt their preferred RO models and communicate with MOSAIC via an API. However, the RAK model can still be consulted in this case as a reference for other kinds of Research Objects, to clarify the requirements and proper usage of the MOSAIC API.

0.3 *MOSAIC Products*

The MOSAIC SDK, as just described, enables developers to create ROs that seamlessly interoperate with MOSAIC portals. Because of MOSAIC's unique design requirements — supporting a rich, multiparadigm semantics for scientific data, and centering a network of self-contained desktop applications — the MOSAIC SDK introduces innovative features that companies may find useful outside the context of academic publishing and of Research Objects. The MOSAIC system prioritizes "Modular Native Design", for applications that are flexible and versatile from a user's point of view, while also securing a software-development methodology that facilitates transparent Requirements Engineering, rigorous code verification, and demonastrable compliance with legal and operational standards throughout the lifetime of a project.

In addition to the SDK, MOSAIC utilizes technologies targeted to both server-side (Cloud) and client-side (Desktop) components. On the server side, NDP Cloud is a framework for deploying cloud services tightly integrated with native applications. NDP Cloud applications can be tested and developed as standalone desktop applications before being uploaded to cloud servers. NDP Cloud also eliminates most of the technological gap between web and desktop implementations: NDP Cloud adopts mostly the same code libraries and programming paradigms as desktop applications (such as Qt/C++), so that client-server interop becomes analogous to networking between two desktop applications. For this reason, NDP Cloud can be a good solution for developers who want to use cloud services to augment the capabilities of desktop software, enabling peer-to-peer messaging and seamless cloud backup and personalization to enhance the desktop experience.

On the client side, VersatileUX is an GUI application framework which leverages the peer-to-peer and personalization capabilities of NDP Cloud. VersatileUX supports a highly modular approach to application development, with tight correlation between data structures and the GUI elements used to display them. Any VersatileUX software is a composite of semi-autonomous modules, each of which is responsible for reading or receiving data conforming to specific structures (from a file or a network) and presenting these data structures in specialized GUIs. GUI and data-structure components are bound together via strong typing. Because VersatileUX components are developed in isolation, testing and compliance verification is simplified for the application as a whole. At the same time, each VersatileUX module can design its own scripting and personalization features, so the overall application can be granularly fine-tuned to the needs of each user.

NDP Cloud and VersatileUX work together to promote personalization and interoperability. Individual desktop applications, built with the aid of VersatileUX, can connect to an NDP Cloud service so as to track user identity across application instances (or even across different applications), while working through the cloud service for data backup and to connect multiple users, for purposes of

messaging and collaboration. VersatileUX modules can be fine-tuned via user preferences and application-specific data obtained from NDP CLOUD services, updating application state through Runtime Reflection. Reflecting MOSAIC's emphasis on complex semantics for scientific data, NDP CLOUD and VersatileUX interoperate according to sophisticated protocols for exchanging strongly-typed data structures between clients and servers and between clients themselves (via cloud messaging). The key difference between this Cloud/Native Hybrid architecture and conventional client/server architecture (including other Cloud/Native paradigms) is that both server- and client-side code takes full advantage of strong typing (selecting the best aspects of Functional and Object-Oriented programming), and a single type system is sustained across all client and server components.

Companies can license the MOSAIC SDK, VersatileUX, and/or NDP CLOUD whether or not they host publications on MOSAIC or host their own instance of a MOSAIC portal.

0.4 *For More information*

Please see the accompanying slides or request a meeting or phone conference to discuss MOSAIC in greater detail!