Clotho 3.0: An Improved Common Framework for Synthetic Biology Computing

S. Paige, P. Vaidyanathan\textsuperscript{1}, M. Bates, J.C. Anderson\textsuperscript{2}, D. Densmore\textsuperscript{1}

\textsuperscript{1}Boston University, Boston, MA  \textsuperscript{2}University of California, Berkeley, CA


tmpaige@gmail.com

**Clotho 3.0 Feature Overview**

**Lightweight Integration API**
- **In the lab**: Use Clotho as a hub to integrate the different software tools in your workflow, without altering their code.
- **At the computer**: Use Clotho as a service in your programs, with as much or as little involvement as you need.

**Server-Side Execution**
- **In the lab**: Run computationally-intensive jobs on your Clotho server through a web interface.
- **At the computer**: Write an app that runs within the Clotho server itself, taking full advantage of the above features.

**Flexible Data Model**
- **In the lab**: Annotate entries with data specific to your research interests.
- **At the computer**: Extend Clotho's built-in data model, or create an entirely new one.

**Managed Format Conversion**
- **In the lab**: Share data between software without format compatibility hassles. Create fluid workflows spanning multiple apps.
- **At the computer**: Aggregate and normalize data from multiple sources.

**Polyglot Programming**
- **In the lab**: Use your existing script tools with Clotho.
- **At the computer**: Write new apps for Clotho in your favorite JVM-supported language.

**Components of Clotho 3.0**

**Imports**
- Clotho 3.0 has import functions for common data formats and databases. These importers are customizable for your own needs, or you can write your own.

**Widgets**
- Widgets display data in the Clotho Web Client. Each widget targets a particular schema, and Clotho’s type conversion logic enables it to display related schemas also.

**Web Client**
- The Clotho Web Client allows users to enter, update, and search data without specialized apps. It provides simple interfaces for authoring or uploading new Clotho components.

**Schemas**
- Schemas describe kinds of data to Clotho, including what information to expect, and how to validate it. Schemas are extensible and customizable. Clotho provides a default model for synthetic biology data, but you can upload your own schemas if your data needs are different.

**Converters**
- Converters change one data format to another. Clotho automatically invokes converters to present scripts and apps with the data formats they are expecting.

**Scripts**
- Clotho can run scripts written in Python, JavaScript, Groovy, and Ruby. Uploaded scripts have access to Clotho’s features through a simple API similar to the integration API. Scripts can be simple functions, or entire feature-rich apps.

**App Infrastructure**
- Clotho 2.0 and Clotho 3.0 both provide basic app concerns: UI, persistence, and security. Clotho 3.0 extends the security model to allow fine-grained read, write, and execute access control over Clotho components.

**App Architecture**
- All Clotho 2.0 apps must be Netbeans Plugins written for the Clotho 2.0 plugin suite. Clotho 3.0 is more flexible: Programmers can use a Clotho instance as a service called from their programs, or write apps designed to run within Clotho.

**Data Model**
- Clotho 2.0’s data model is rigid and prescriptive, ensuring that any data entered conforms to a fixed set of schemas. Clotho 3.0’s data model accepts extension of any schemas with additional data, and allows users to define their own schemas.

**Sample Workflows**

**Add Data and Script for Reuse**
- Once your data and script are in Clotho, they are integrated with Clotho’s features. You can run the script from Clotho’s web interface, save the results in the database, and incorporate your script into a larger app.

**Aggregate Local Inventory and Remote Registry Data**
- Using the web client, you can aggregate the Registry information with your lab’s logistical storage info, discovering which parts are already available for use, and save the aggregated data in a normalized format your other tools can use.

**Generate Designs Using Published Parts**
- This workflow generates candidate device designs and generates plans for their creation by wiring together a parts registry, EugeneLab, and Raven.

---

Icons adapted from aws.amazon.com/architecture/icons. We thank Swapnil Bhatia, Aaron Neukroth, and Jenhan Tao for discussion, critique, and code contributions.