

Clotho 3.0 Feature Overview

Lightweight Integration API

In the lab: Use Clotho as a hub to integrate the different software tools in your lab, 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.

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.

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.

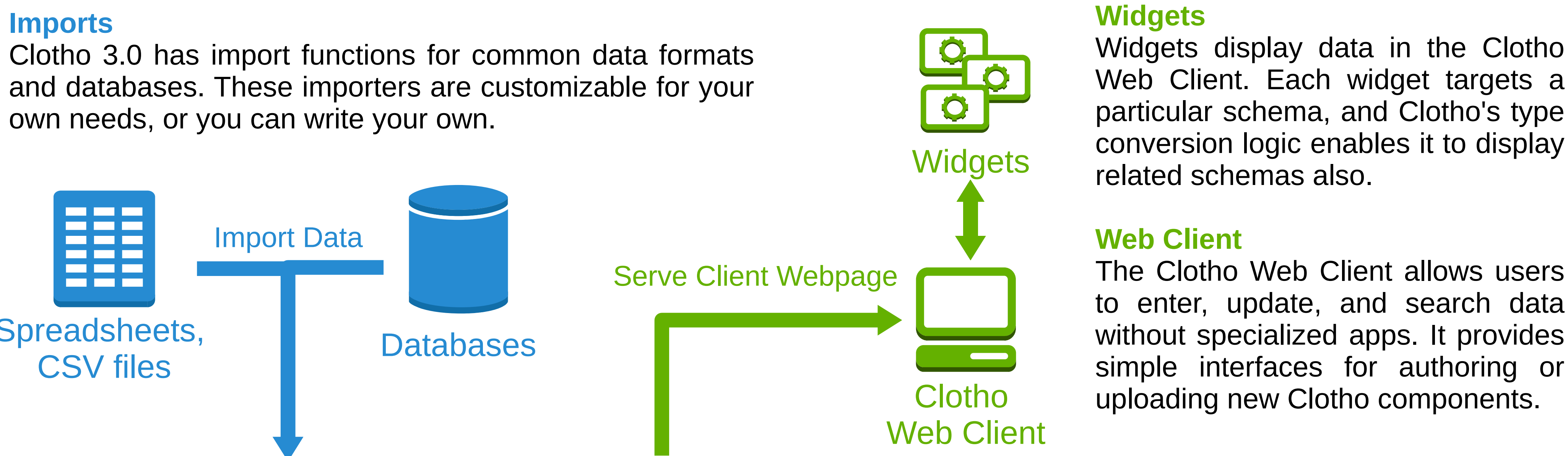
Managed Format Conversion

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.

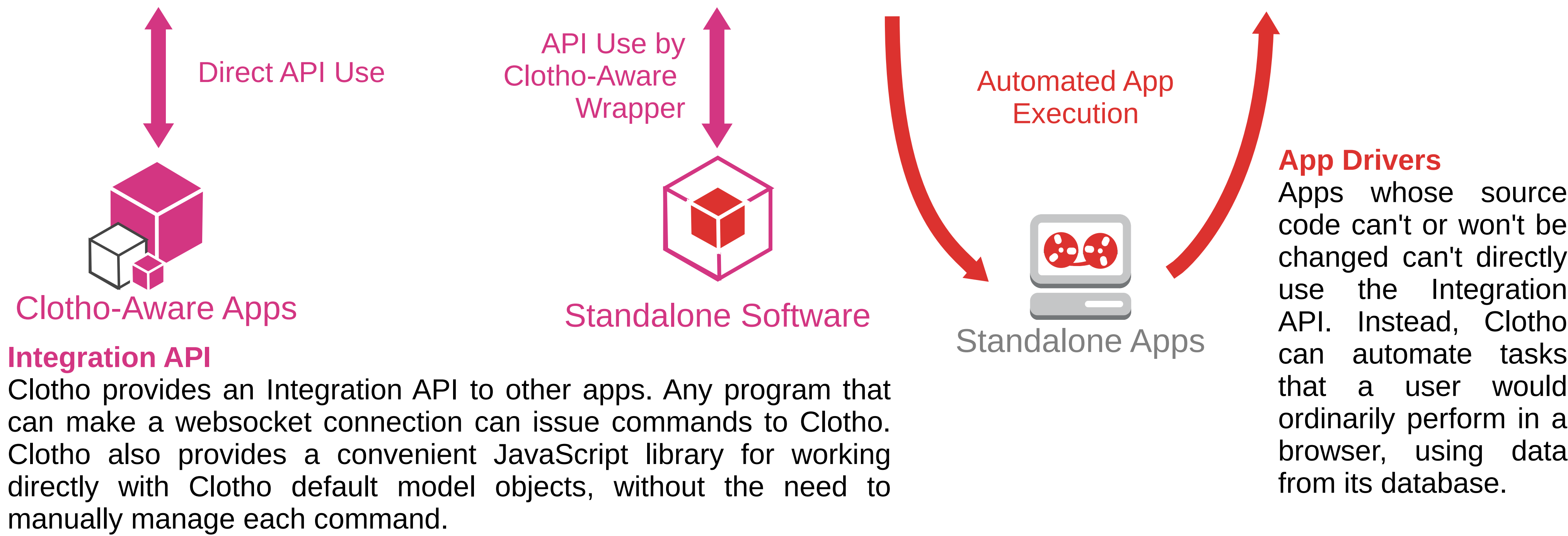
Components of Clotho 3.0



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.



Clotho 2.0 vs. Clotho 3.0

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.

Community Integration

The default data model in Clotho 3.0 improves on Clotho 2.0's community-designed model, incorporating user feedback and adopting community standards such as SBOL.

Script Compatibility

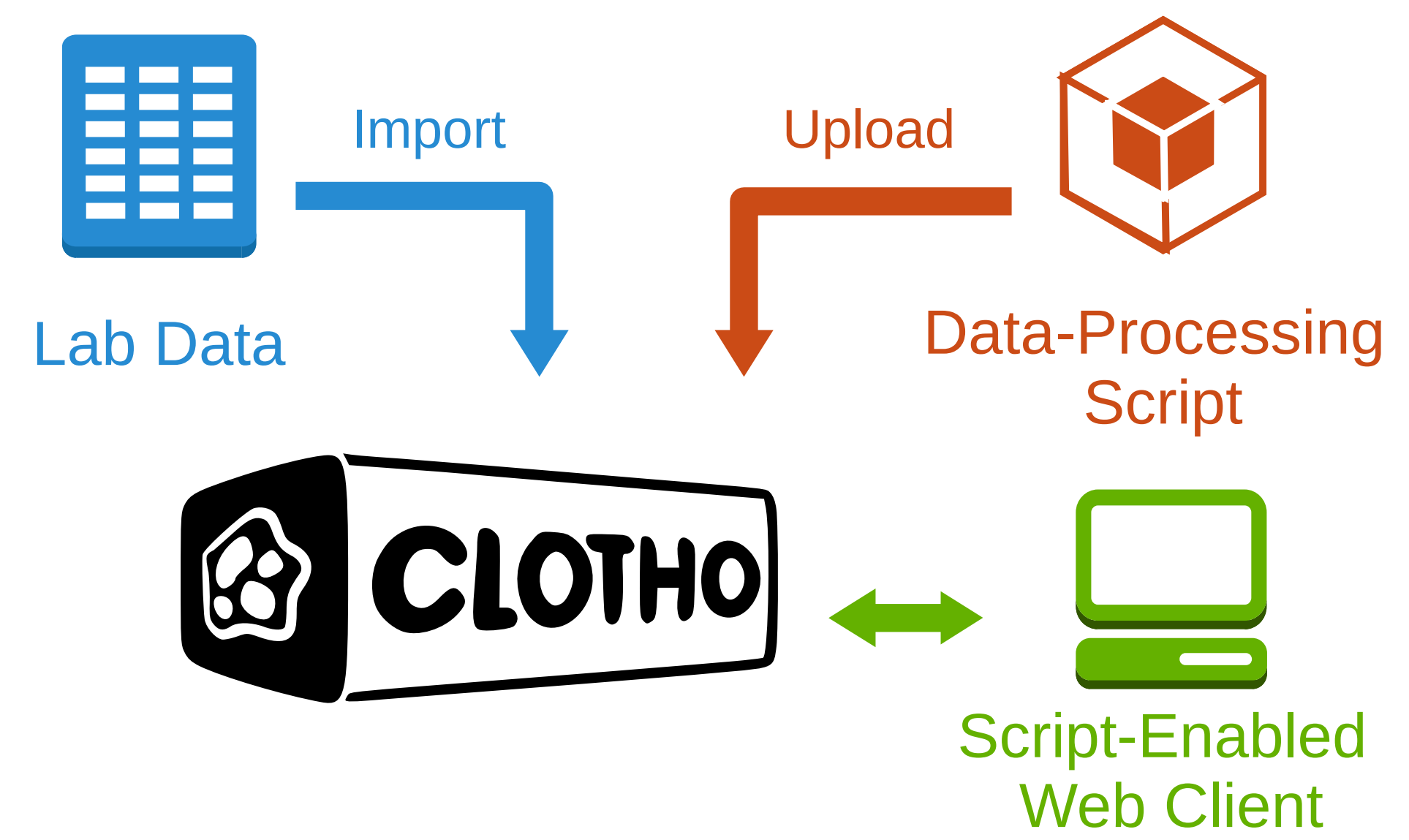
Previously, all Clotho 2.0 development had to occur in Java. Clotho 3.0 is compatible with scripts written in Python, Ruby, and JavaScript, as well as Java itself (both traditional Java classes and scripts written in Groovy). Tools written in different scripting languages can be combined inside a Clotho app.

Technology Stack

Clotho 2.0 uses a heavy Java client backed by an SQL database. Clotho 3.0 provides an HTML5-based web client backed by a document store database. 3.0 extends the range of technologies accessible to Clotho by providing a websocket interface for programmatic access to the integration API.

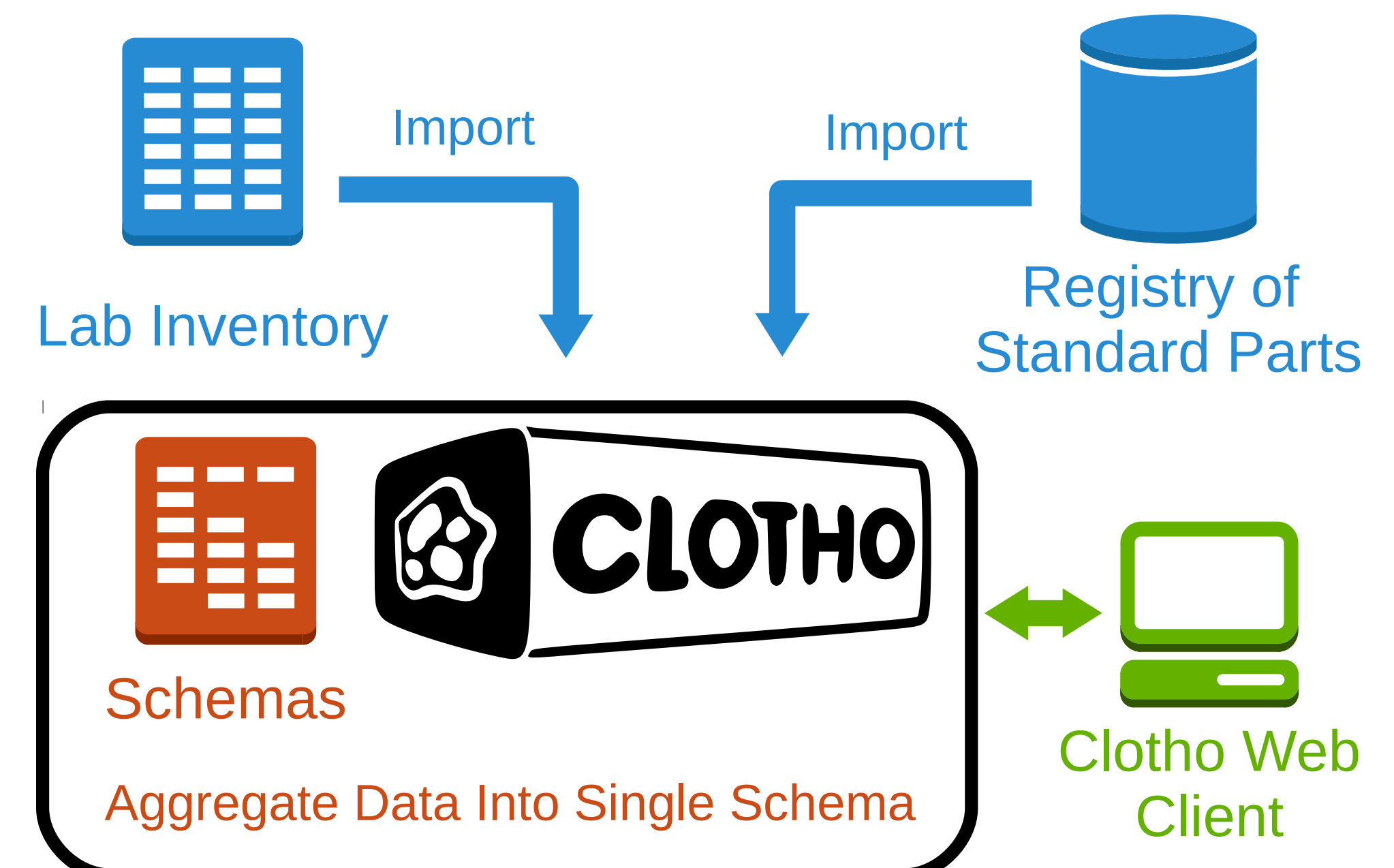
Sample Workflows

Add Data and Script for Reuse



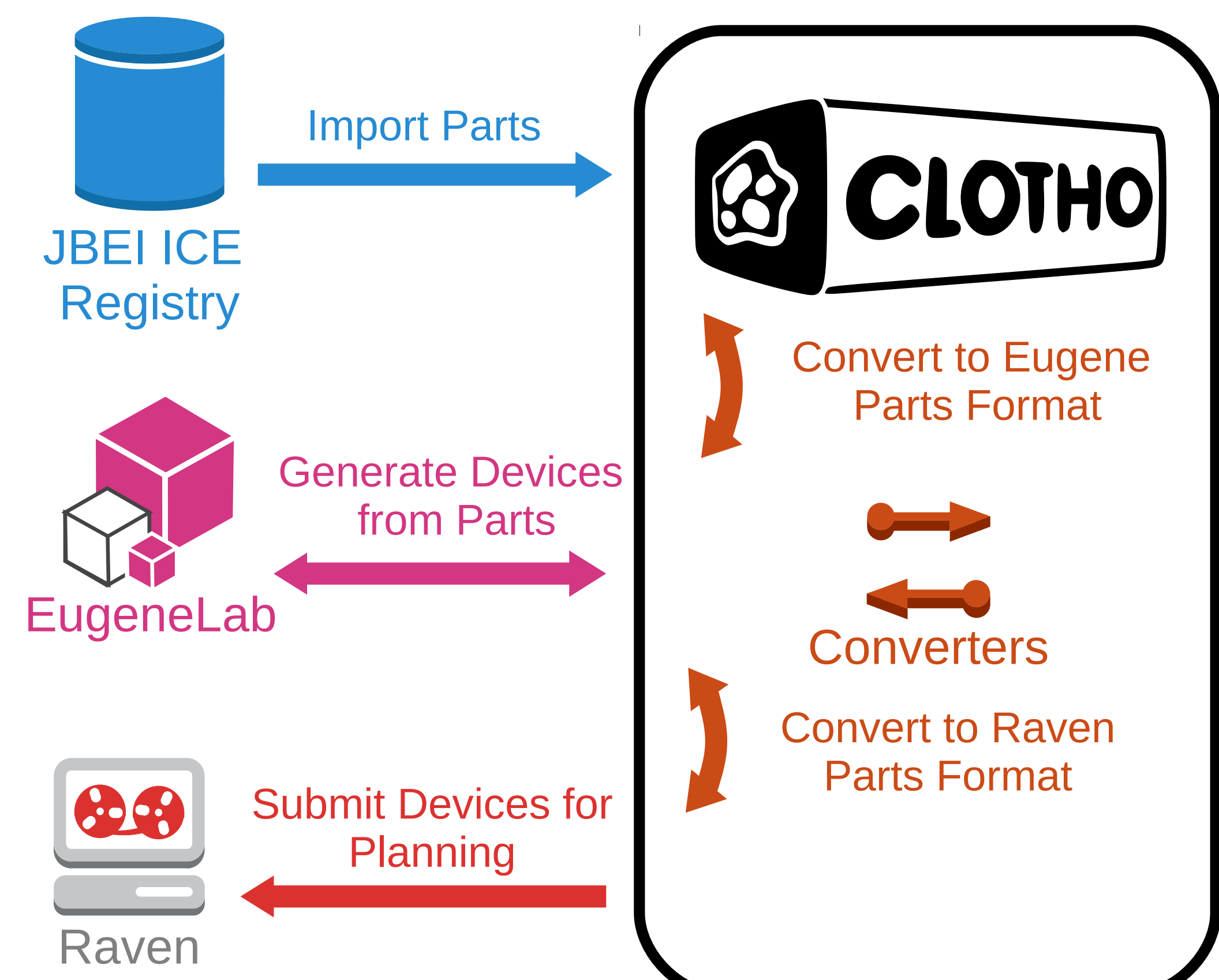
Once your data and script are in Clotho, they are integrated with Clotho's features. You can run the script from Clotho's web interface, save the results in the database, and incorporate your script into a larger app.

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.