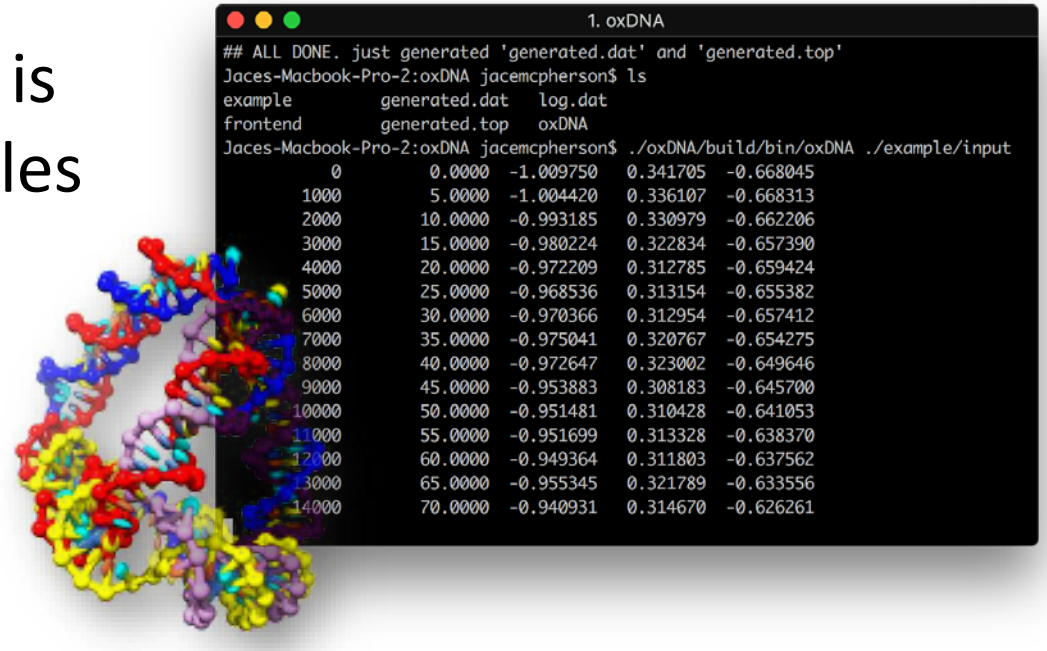
WebDNA Project Proposal

Jace McPherson, Salvador Sanchez, Zack Fravel,
David Darling, Jonathon Raney

November 6, 2017

Premise: Here's the Problem…

- oxDNA is a widely used DNA simulation software.
- It's clunky, and very inaccessible to most DNA researchers.
- Analysis of the oxDNA output is the main time-consumer for DNA related research.
- Because it lacks a UI, file management is also an issue. oxDNA produces many files and it's up to the user to figure out how to organize them.
- Difficult visualization of simulation results



Objective: How can we fix this?

- We plan to wrap oxDNA functionality with a website
- All input data, simulation visualization, and data analysis should take place in our controlled environment
- Guide the configuration experience with a clean, sensible UI
- Allow advanced users to download their results and perform offline analysis, if necessary.

Design Goals

Technologies

Server

Django Python Server

- Listens for HTTPS requests
- Serves web pages using the Django HTML renderer

PostgreSQL DBMS

- Widely used SQL extension

Client

Angular2 Renderer

- Framework based on Javascript for rapid development of UIs.
- Adapts and extends HTML to present fully dynamic content.

Three.JS

- 3D Renderer for browsers (using WebGL)

Project Schedule

Tasks have been divided into two main sections to be worked on concurrently

Server

- Setup of hosting method — January 26
- Initialization of database schema on hosted PSQL platform — January 26
- Server Endpoints — February 14
  - POST user authentication
  - GET projects
  - GET project configuration data
  - PUT project configuration data
  - POST specify simulation generation scripts
  - POST specify analysis pipeline
- Implementation of file generation — February 16
- Script uploading and checking — February 28
- Execute simulation from project data — March 16

Client

- Login/Registration page — January 22
- Project management page — January 30
- Configuration page — February 7
- Output file visualizer — March 9
- Output pipeline editor — March 23
- Custom script manager — March 30
- Dashboard (integration of pages) — April 6

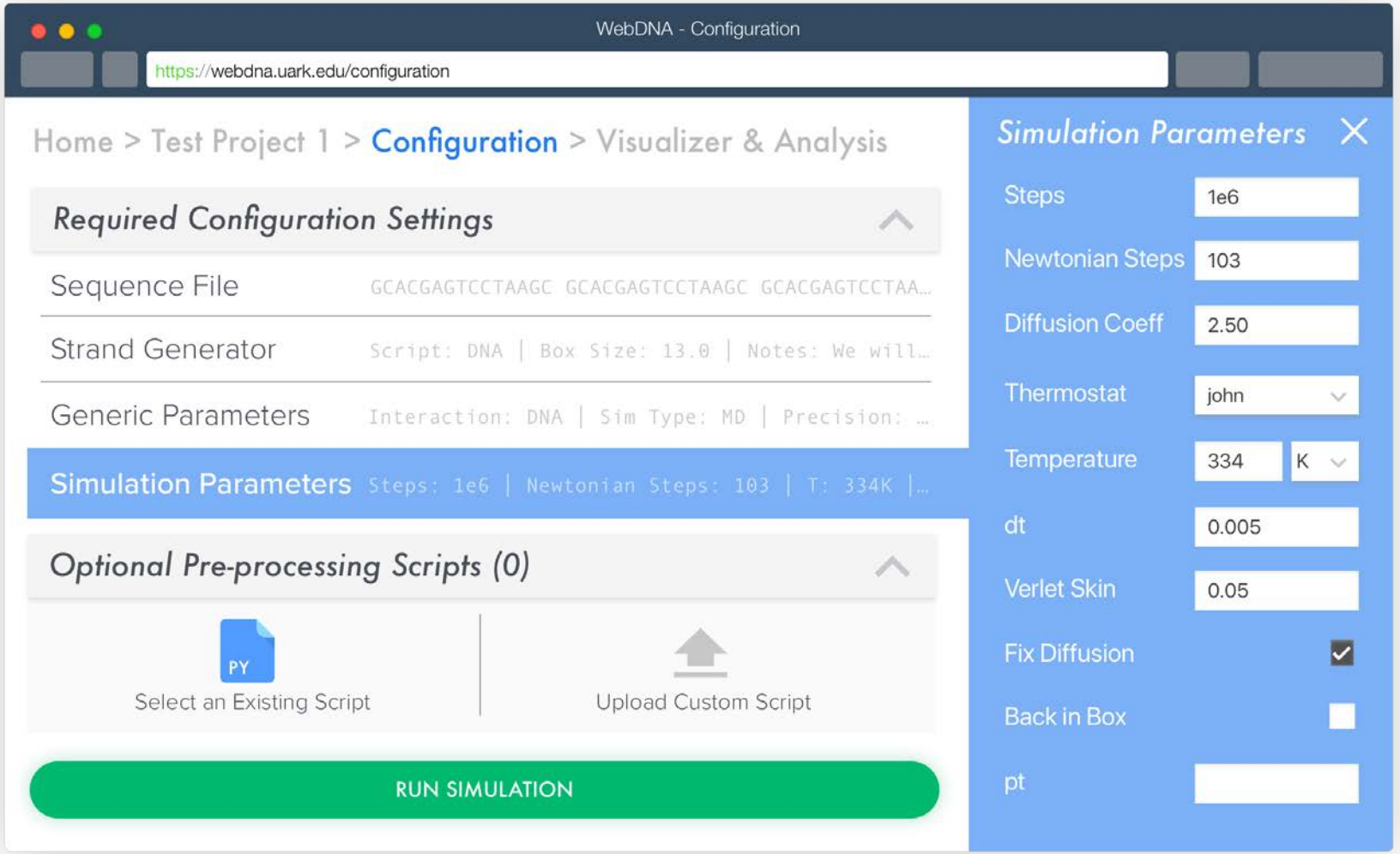
Client: Simulation Configuration

Required Settings

- Normally set up with Python scripts in a clunky workflow
- Now, reduced to simple series of steps with *same functionality*

Optional Settings

- Allows users to run built-in and/or custom scripts on their input



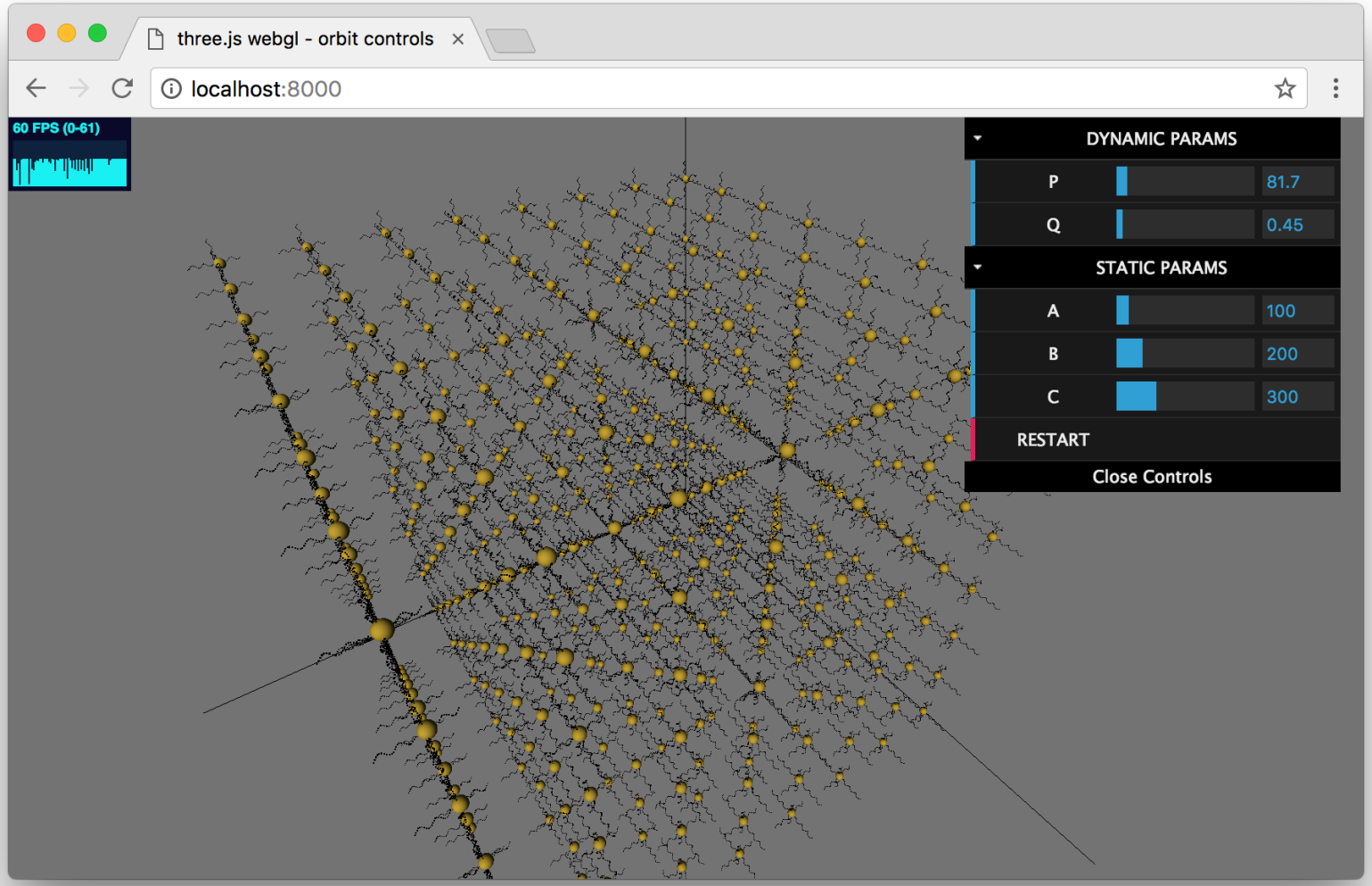
Client: Simulation Visualization

Current Visualizer

- Simple nanoparticle visualizer already implemented
- We can use the current visualizer as a base for our more flexible oxDNA simulation visualizer.

Added Features

- Scrubbing through simulation over time



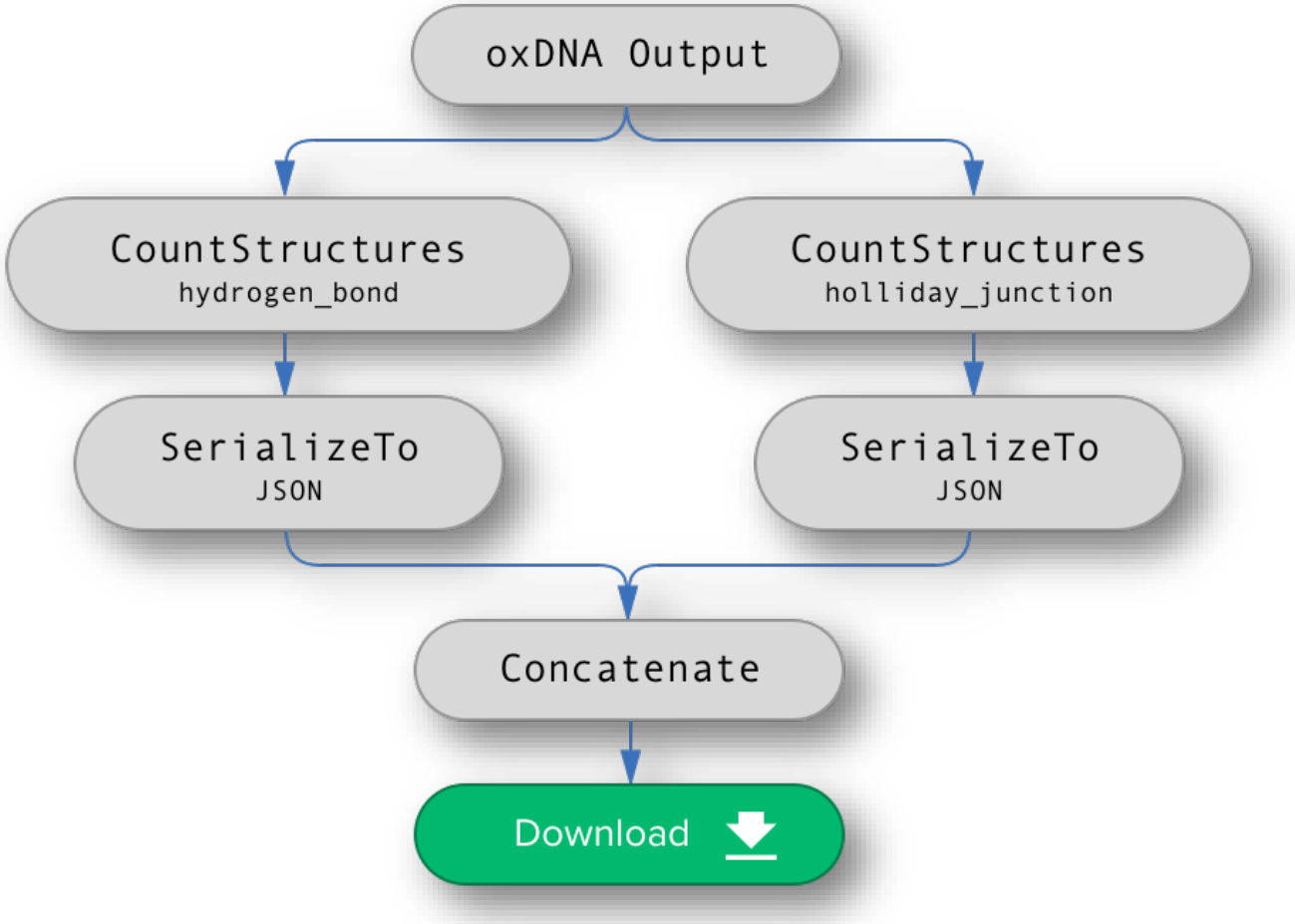
Client: Output Pipeline Editor

Current Analysis Pipeline

- NONE! Researchers have to write their own analysis code to extract meaningful data from simulations

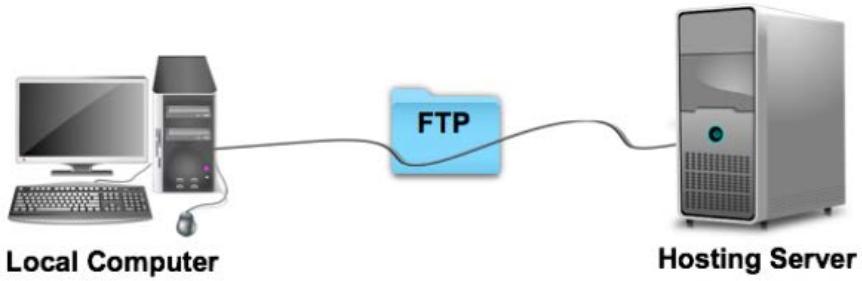
Our Analysis Solution

- Create a drag-and-drop style interface for users to pipe their simulation data through a series of analysis steps
- Allow downloads of data at *each of the steps*, including the final output, for full customizability.



Client: Custom Script Manager

- The goal is to provide a simple way for users to upload and make use of their own locally-written python scripts for use within the pipeline editor.
- A remote file exchange protocol such as FTP will be used to allow users to directly upload files
- These files will be screened to ensure they meet server requirements and then made available to users



File Transfer Protocol

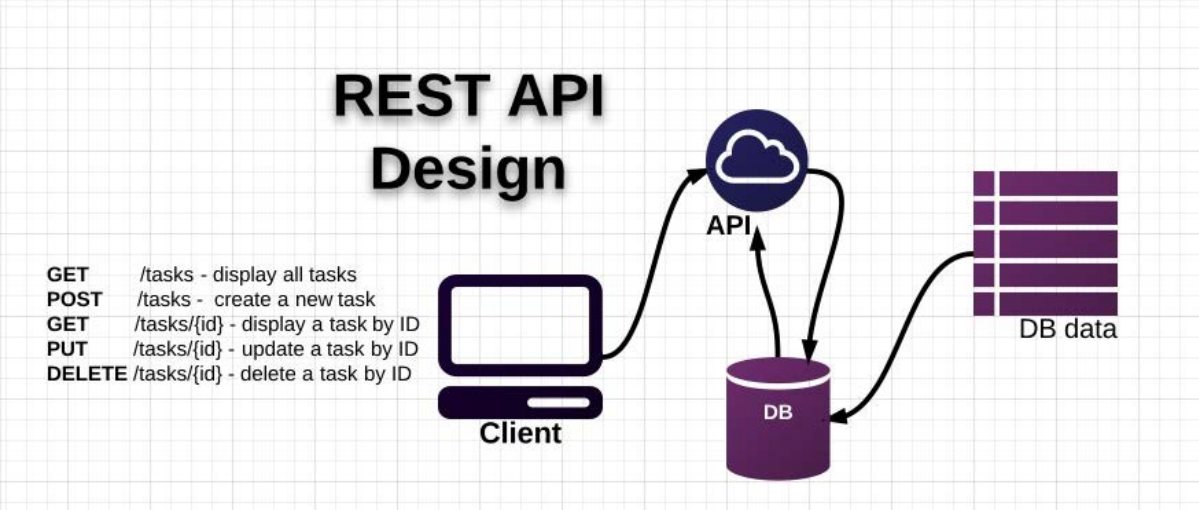
Server: Hosting and Database

- Server will be hosted on either UARK's Turing or a 3rd party platform such as AWS or Heroku

- The PSQL Database will be similarly hosted and will store data such as user information and script metadata

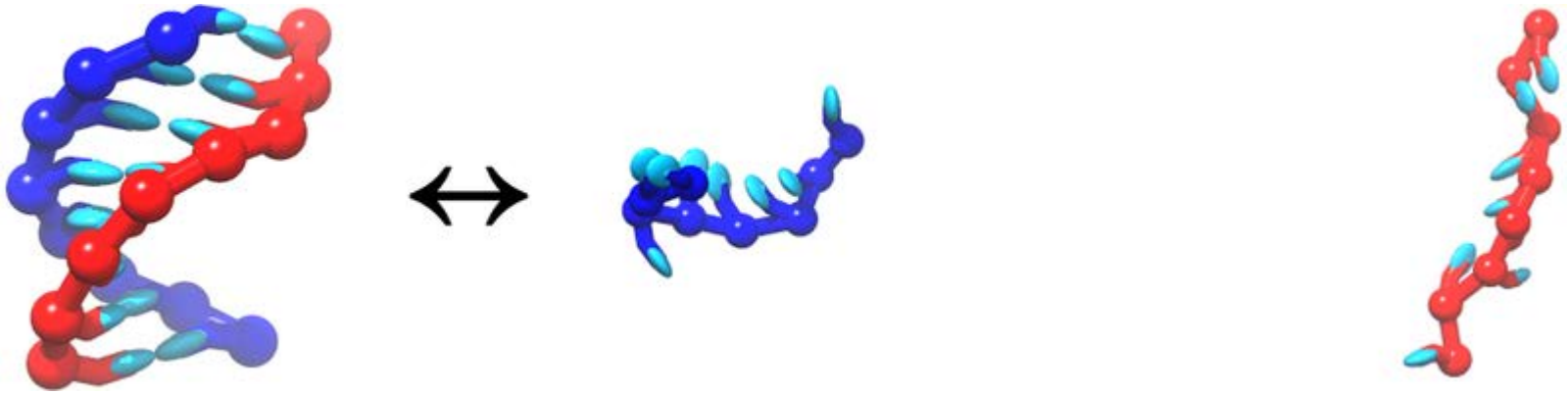
Server: URL Endpoints

- The server will rely on RESTful URL endpoints to provide functionality to clients.
- Endpoints will be defined to perform many different tasks such as authenticating users and specifying analysis pipeline information



Server: Executing Simulations

- The user will be able to submit a Job for queuing to the server
- The Job is initialized using only the desired project ID.
- An instance of oxDNA will be executed according to the parameters specified in the project corresponding to the supplied project ID.



Final Product

- By the end of the course, we hope to deliver a robust and extensible web server capable of launching highly customizable oxDNA simulations.
- 3D analysis tools and custom output scripts will offer useful and intuitive functionality for researchers.
- Using oxDNA will become much more streamlined and simplified