SPHERE-E Web Architecture
Sphere-E demo

- Show working set usage
- Show scenario usage
Underlying Data

- Given data on products
  - Product information
  - LCA (environmental impact) data
  - As a MYSQL database dump
  - At 3-6 month intervals
Application Data

- Kept in a SQL database
- Queried and updated as needed
- About 40 tables
  - Initialization script
  - Update script
Boilerplate Files

- **Config.js**
  - Configuration information
  - Basic language extensions
- **Log.js**
- **Email.js**
- **Database.js**
- **Initialize.js**
Front vs. Back End

- Do as much as possible front-end
  - All data is too much
  - Data for one type of product is okay
- Sorting, filtering, selections
  - All done in the front end
  - Values saved by the back end
- New page (consider vs evaluation ...)
  - Done in the back end
Authentication

- Auth.js
- Roles and permissions
  - Store as relations in the database
- Sessions
- Login, registration
- Forget login, forget password
- Display user settings
Data Management

- Data.js
  - Read the whole database in from SQL
  - Convert to internal structures
- Get all categories
- Get all products for a category
  - Limit by geography
- Compute benchmarks
Static Pages

- General.js
- FAQ, about pages, home, terms
- Compute FAQ html from simpler form
Project Management

- Project.js
- Middleware for project commands
- Project home page
- New project, edit project
- Delete project
- Manage project users and roles
Working Sets

- Workingset.js
- Middleware for workingset commands
- Display working set
  - Consideration, evaluation, cost, generation pages
- Edit working set
  - Set selections, filters, sort order
  - Set properties (amounts, etc)
  - Set product properties (costs)
  - Add comments
Scenarios

- `Scenario.js`
- Single page for scenario
  - Handle all updates in the front end
- Setting properties
  - Costs, amounts, importance
Server

- Server.js
- Express-based dispatch
- Lots of error handling
- Forever to keep it up