JavaScript: Bringing Object-Level Security to the Browser

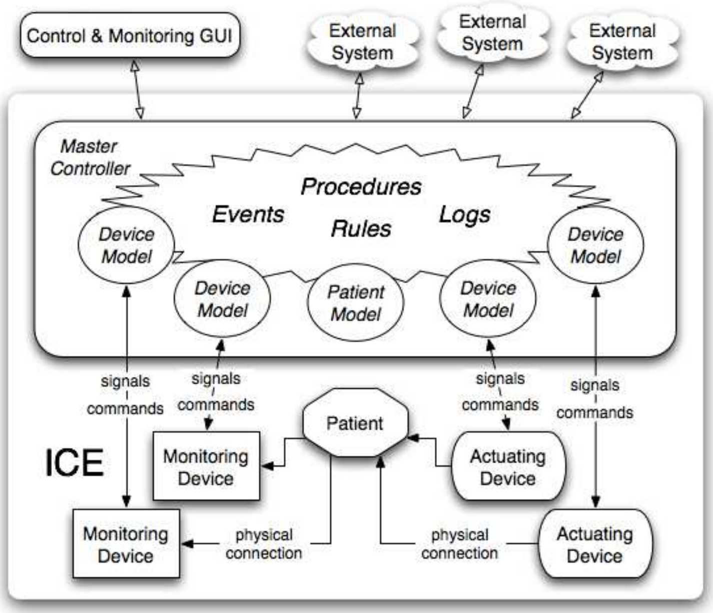
Charlie Meyer

Problem Recap

- JavaScript is the world's most popular programming language
- JavaScript is a dynamic language, data can be changed at run time by anyone
- Many pages load scripts from multiple sources
  - Every script has the ability to change any piece of data
- Security does not receive sufficient attention due to
  - The complexity of web based applications
  - The ad hoc process of development
  - Many web designers do not have the necessary security knowledge on web development techniques

Results

- Created HotSausage JavaScript Framework, which includes a module for enabling object-level security on data
- Applied framework to existing medical device simulation to gauge effectiveness, overhead, and usability



Framework Operation

- Enable privacy on an object
  - Gives access to a “purse”
- Put private data in purse
- Write functions that access private data
- Add those functions to objects using framework methods
- Future calls to those privileged methods are invoked through the framework to ensure privacy and integrity

Analysis

- Discovered several new vulnerabilities, behavior testing was crucial
  - Fixes have been designed for some, implementation in progress
  - Others cannot be addressed via implementation
- Initially thought processing overhead would be a major factor, but real world use proved that not to be an issue

Conclusions and Further Work

- The framework does fulfill the goals that we initially laid out
  - Object-level privacy
  - Data integrity
  - Low memory footprint
- Case study proved valuable
- Integrate framework modules with CommonJS
- Planning to release framework publically in 2010