Technology

Language:

- Java
- Javascript

Backend:

- NodeJS / WebAPI
- MongoDB + GridFS
- ES
- Ningx
- DFS
- Docker
- Big data - Map-reduce (-)

Front-End:

- AngularJS/Angular2
- JQuery (-)
- D3 (-)

• Product Core is using WebAPI. But suggest to use WebSocket or similar technology (e.g. SocketIO) to set up the dual connection between client and server, meanwhile use message bus/queue to enable the communication.

• WebSocket requires Windows 8/Win Server 2012 or higher version. Neither does it work with earlier version of IE.

• Product uses local cache (Sinopia) server for 3rd parties but would like to have global server, too. It seems Taiji use some local cache, too.

• Product supports Copy/Paste deployment, so far only do smoke test on continuous deployment

• Product uses gulp for build, but WebPack may be a better choice especially for managing external modules

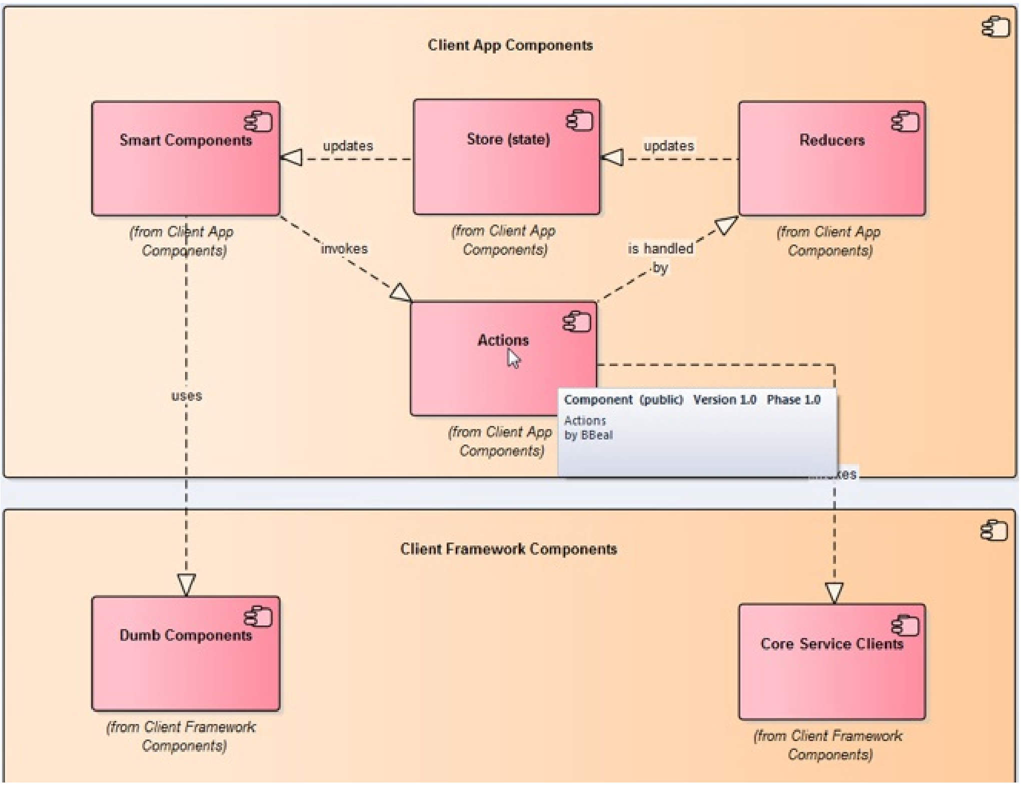
• Product will drop JSPM since it's not keep updating and some functions are not properly defined

• Brandt suggests for common web Uis like plots/gauges, using INT may not be necessary. But for domain display, like WellLog, etc, INT is a good choice, they have improved a lot recently. From Eric, we're setting up the communication channel with INT.

• Product support latest IE and Chrome, with basic workflow support in Safari. Most of the users will use Chrome thought it is not SLB IT standard yet.

• Product uses immutable model (Flux=>Redux/Rx? Wrappers in ng include ngrx, ng-redux, ng2-redux), and model is stored as state tree which is very easy to restore.

• Product uses Redux on client side to store the model(state), Smart components updates according to the stated model. Angular2 is used inside SmartComponent.

- 

Client App Components
Smart Components
updates
Store (state)
updates
Reducers
(from Client App Components)
(from Client App Components)
(from Client App Components)
invokes
is handled by
Actions
uses
(from Client App Components)
Component (public) Version 1.0 Phase 1.0
Actions
by BBeal
...xes
Client Framework Components
Dumb Components
Core Service Clients
(from Client Framework Components)
(from Client Framework Components)

- Product is hosted in IIS, which provides functions like DOS which are not available in self-hosted. Another approach is to use NGINX.
- Both client and server are deployed on the same machine. Need config IIS after installing kit, web server is always on
- Product this year will try to adapt Angular2 + Redux, and integration them with WebWorker to achieve better performance. See roblog.io.