iSIMLab

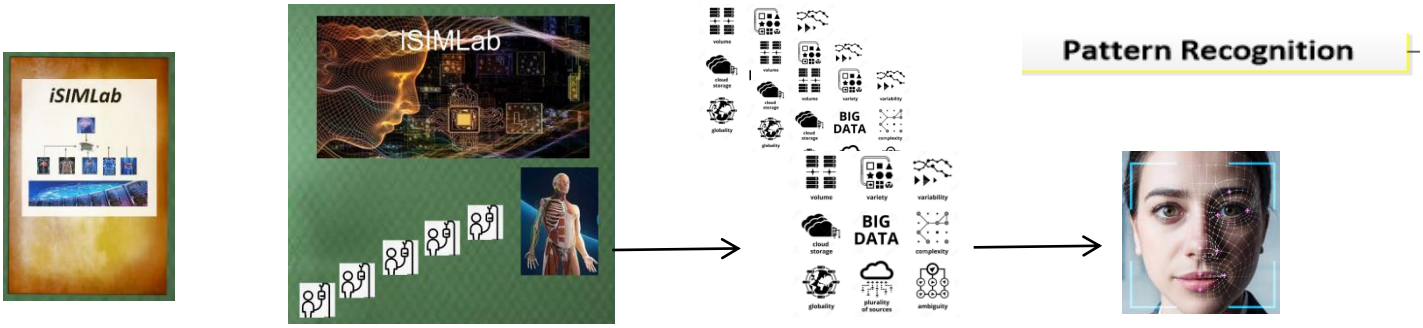
A simple[1] but effective approch

Alias : A Basic Description v 1.0

iSIMLab's Concept

iSIMLab is an Internet simulation laboratory where

- Large complex systems can be built quickly
- Each complex system can be cloned with a small code change
- Each clone ou tput is auto-compared against one another
- Big output data would be generated awaiting pattern recognition



iSIMLab's original purpose was to

- Create a human body simulator in the area of cancer research
    - iSIMLab would build a communication infrastructure (IF) between modules
        - eg. Lungs, Liver, Pancreas, ...
        - the IF would create & run a system loopback POST[2] msg verifying all module message/command paths
        - the application would then run all 'Init' prefixed commands
        - go into a main loop running
            - a cancer progression
            - an IV treatment

[1] Non-conventional
[2] Power-On-Self-Test

iSIMLab Design

iSIMLab's design is intended to facilitate users[3] by

- building the communication, POST and appllication plugin infrastructure
- by using a small set of cfg'ation files
- users will normally copy a project w/these cfg files
    - making a small set of changes developing their unique application

Configuration files

- can be large and as users normally change only a small set of values
- iSIMLab uses a default cfg'ation placeholder file from which the users sets a small number of properties

The research user defines a small set of simple scripting files

- hiding the application programming language complexities

Researcher	Copies & edits Script files	iSIMLabGenerates cfg files	used to build system & create clones

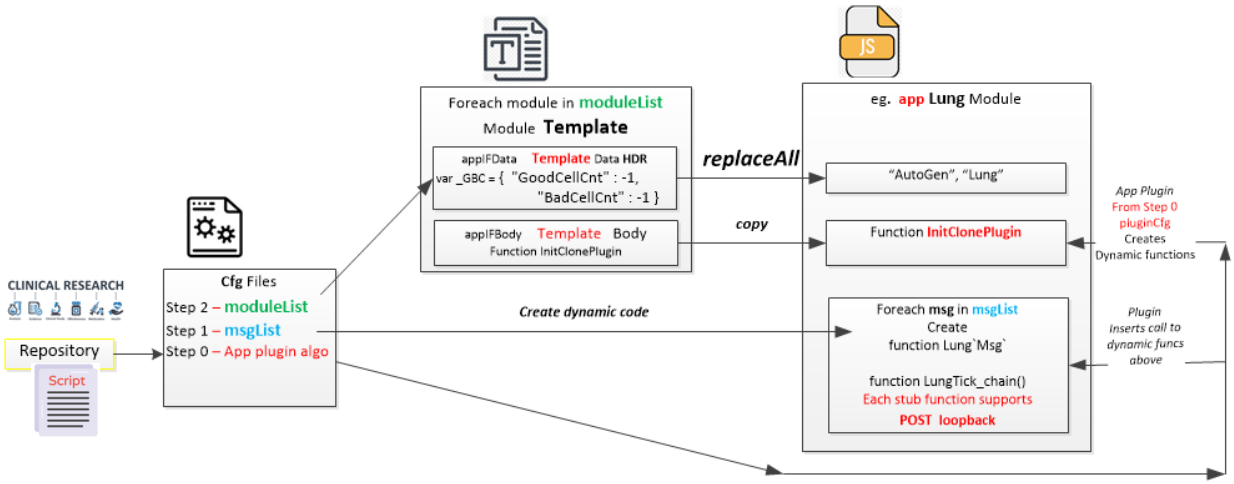
[3] Cancer researchers to build simulation projects (and doctors to access the projects)

iSIMLab Implementation

Build and run speed is what separates iSIMLab.

iSIMLab is implemented via an intepreted language

- saving the compile build & link times of classic languages
- allowing application dynamic algorithms to be plugged-in at runtime
- test coverage is accomplished by system POST msgs and custom utillities



iSIMLab is a program that builds a program by

- utilizing templates
- using cfg file msg and module lists to build the source code
- embedding the communication IF, POST and plugin features
- where at runtime application code injection into the plugin architecture is achieved
    - runtime plugins can be imported from medical repositories

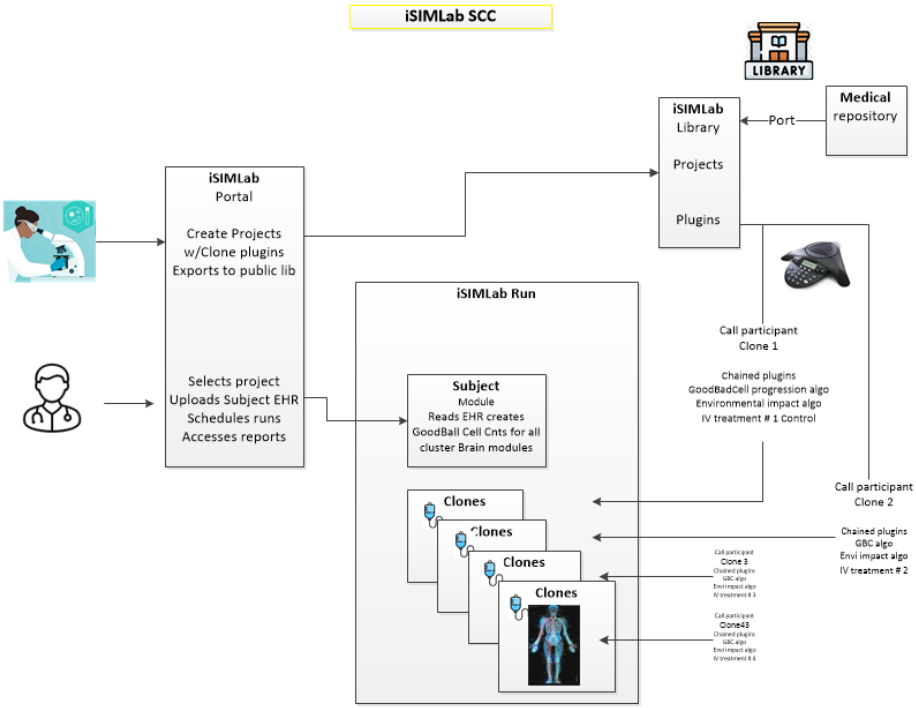
iSIMLab's goal is to build a 10,000 module system in less than 1hour.

iSIMLab Analogy

iSIMLab was created with a telephone conference call in mind. In fact iSIMLab's previous name was SCC. Short for software conference call.

The basic idea was for iSIMLab to be used as follows:

- a doctor connects to iSIMLab
    - Enters a non-metastatic lung cancer stage III patient request
    - iSIMLab connects to medical repository avatars around the world where
    - case discussion follows

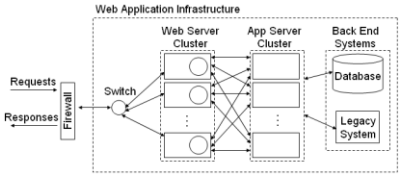


The specific idea for iSIMLab

- a researcher connects to iSIMLab & builds an iSIMLab Lung cancer project
- a doctor connects to iSIMLab and
    - chooses a iSIMLab Lung cancer project
    - enters subjects EHR
    - iSIMLab builds a Good/Bad cell model of the subject
    - clones that subject and runs a different treatment protocol on each clone
    - at end of the run the Prognosis module is called where
        - the Good/Bad cell counts of each clone are used to generate
        - a comparision matrix comparing the 5 year survival rate and quality of life
- SCC is illustrated at the runtime by
    - Programed phone call pulls module participants from worldwide repositories
    - Each module participant responds 'here' to the endaround POST msg
    - The agenda is programmed into the main loop where
        - 'Init' msgs/commands are executed initializing the system
        - The cfg'ation files main loop is entered where
            - A unique IV treatment is plugged into each clone
                - IV Plugins originated/pulled from medical repositories

Scalable

iSIMLab considered super scalar computers as the run environment but chose a less expensive but effective eCommerce cluster server approach.



iSIMLab distributes load across many server clusters allowing near real time execution.