

*About **NASA**, **LUA**, and **Beer***

Jelle Ferwerda

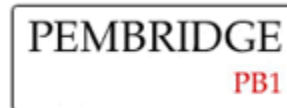
Co-authored with Ashwin Hirschi

Who We Are & What We Do


- Founded in 2000
- Located in Washington, DC
- Operate globally

- Software development environment for *business experts*
- Business experts make software applications unique to their organizations

Who Uses It



LogicNets System

 Modules

 Account Settings

Sat, Jul 12 2008

Jelle Ferwerda - LogicNets



[designer](#)

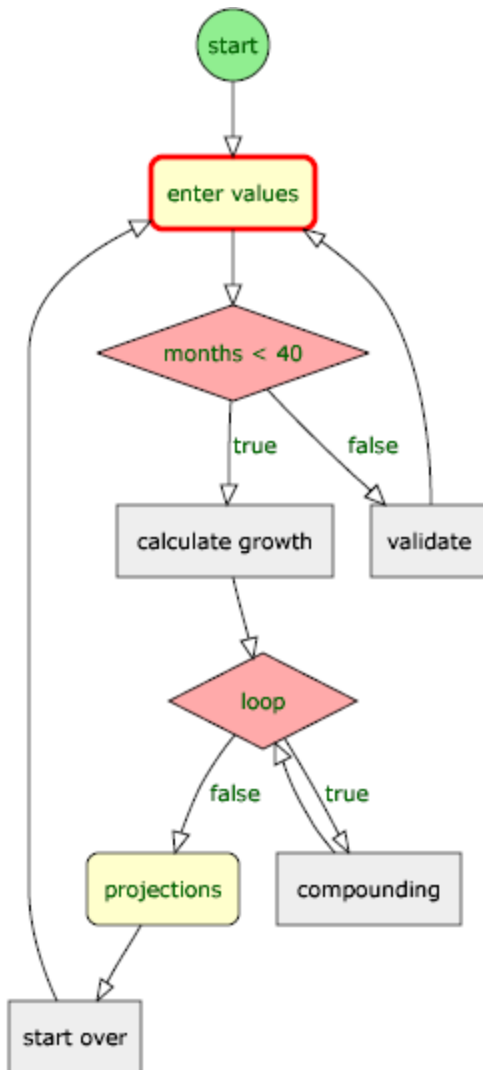


[publisher](#)



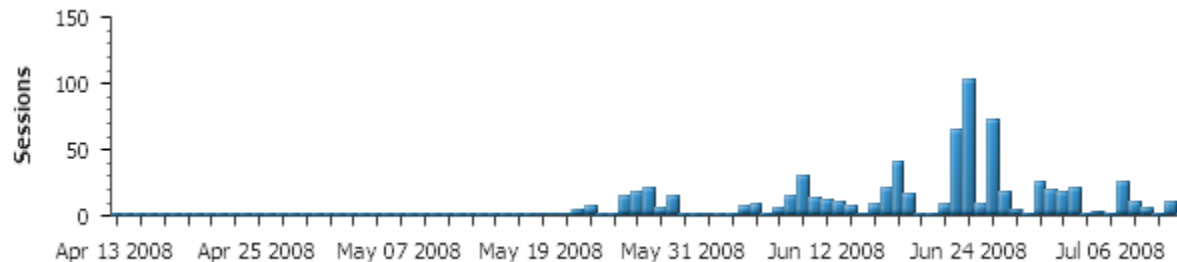
[help](#)

Designer



- Target audience: Business experts!
- Define applications as logicnets
- Nodes represent:
 - Forms
 - Rules / Routers
 - Call nodes
 - Process (server based actions)

- Publish as web applications
- Track usage
- Collect feedback
- Publish as stand-alone applications

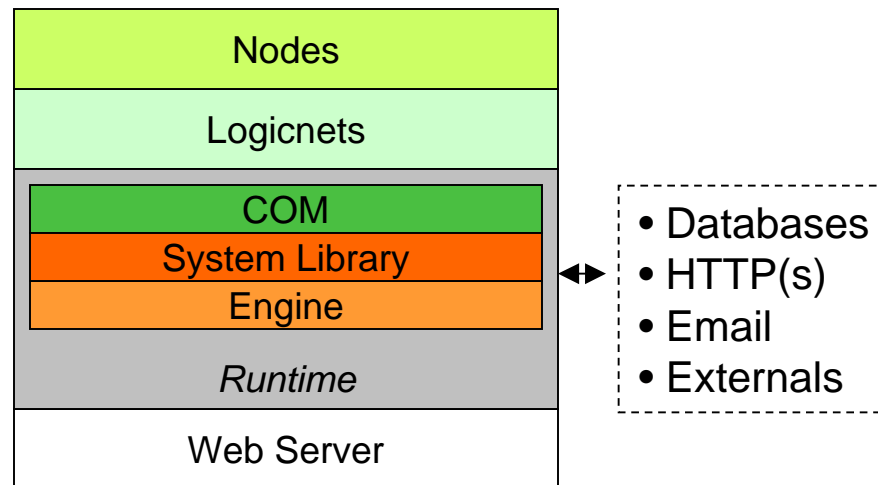


Implementation Requirements

- Performance
- Scalability
- Stability
- Flexibility
 - Easy to add new functionality
 - Ability to work with a dispersed team

First Implementation

2000 - 2004

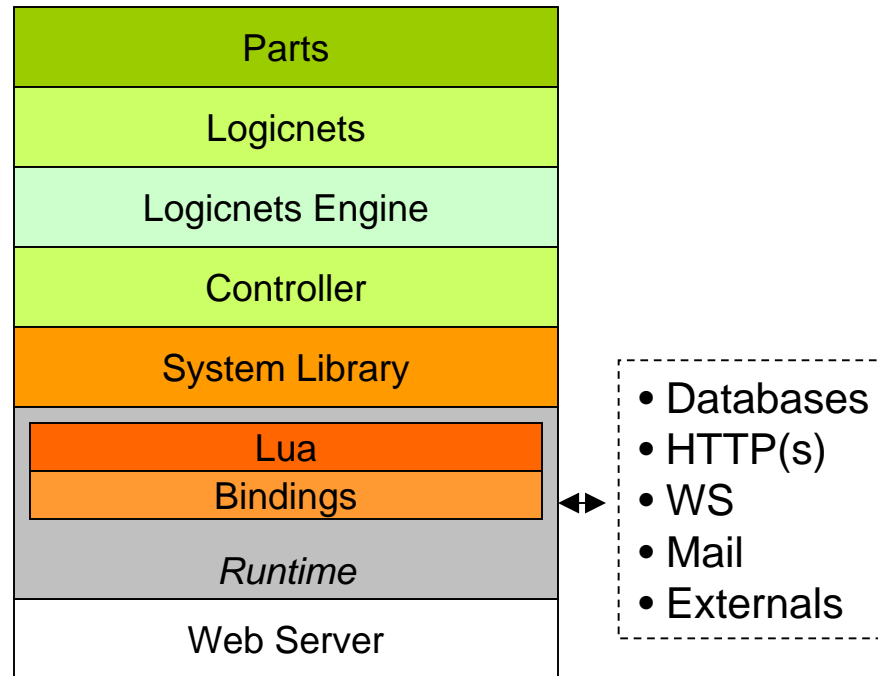


Over time:

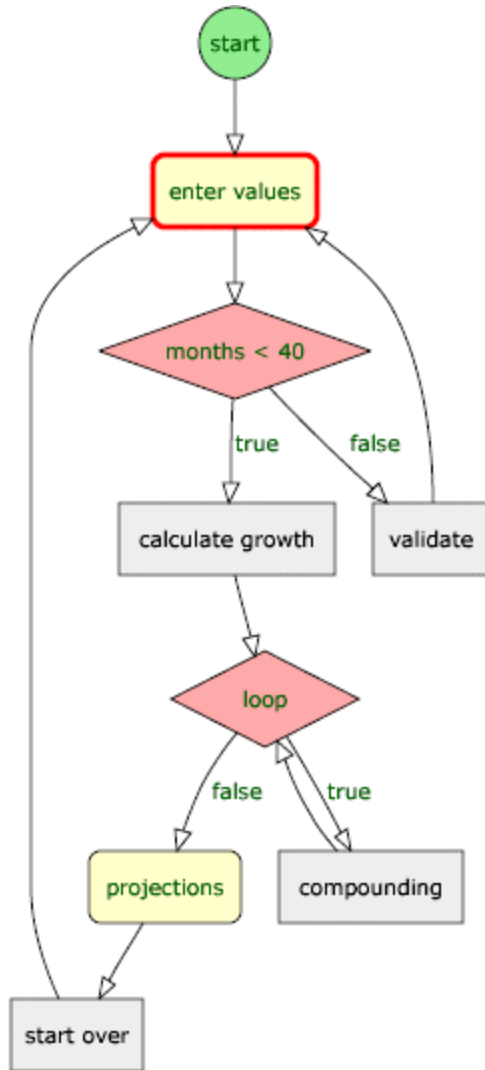
- Flexibility...engine is too static, node level is too generic
- Performance...more customers leads to bad performance
- Stability...not as stable as necessary

The Lua Implementation

2004 - Now



Logicnets in Lua



- Each Logicnet is translated into Lua tables containing the Net and Node properties
- Each Node contains composition of parts defined as Lua tables
- Each part's behavior is mapped to corresponding Lua script

Form Node Editor UI

The screenshot displays the LogicNets Form Node Editor interface. At the top, there are tabs for 'Details', 'Form', 'Elements', 'Links', 'Notes', and 'Feedback'. The 'Form' tab is active, and the title bar reads 'Form Editor - node 10'. Below the tabs is a rich text editor toolbar with various icons for text formatting, alignment, and insertion. The main content area is titled 'Revenue Growth Calculator' and contains the following text:

You have created the perfect widget, and you want to sell that widget to the world. You will charge your customers a monthly fee to use this widget.

To determine the amount of money you can earn selling your widgets over a period of time, simply enter the number of widgets you plan to sell in your first month of sales. Then enter the amount customers must pay monthly to use your widgets.

Enter the percentage amount of growth in sales you estimate each month you sell your widgets. Finally, add the number of months you plan on selling these wonderful widgets.

Below the text is a form with four input fields:

Starting Sales:		<input type="text"/>
Monthly Sales Charge:	\$	<input type="text"/>
Growth Rate:	%	<input type="text"/>
Months to Sell:		<input type="text"/>

At the bottom of the form is a 'Calculate' button.

Form Node Lua Translation

```
node {
  can_start = 1,
  outputs = {
    {
      target = 70
    }
  },
  parts = {
    "<P><STRONG>Revenue Growth&nbsp;&nbsp;&nbsp;Calculator</STRONG></P>\n<BLOCKQUOTE dir=lt:
    {
      _name = "table_test"
    },
    "<TBODY>\n<TR>\n<TD>Starting Sales:</TD>\n<TD></TD>\n<TD>",
    {
      type = "text",
      data_type = "numeric",
      _name = "text_input",
      path = "res.growth",
      location = "res",
      validation_text = "Please enter a number in this field.",
      var = "growth",
      default = "10",
      validation = "mandatory"
    },
    "</TD></TR>\n<TR>\n<TD>Monthly Sales Charge:</TD>\n<TD>$</TD>\n<TD>",
    {
      type = "text",
      data_type = "numeric",
      _name = "text_input",
      path = "price",
      location = "",
      validation_text = "Please enter a number in this field.",
      var = "price",
      default = "",
      validation = "mandatory"
    }
  }
}
```

Advantages

- **Flexibility**
 - Many previous core services are now dynamic, including the engine
 - One part relates to one Lua script
- **Performance**
 - Much(!) faster scripts
 - No XML overhead
 - Runs as cgi implementation
- **Stability**
 - A different world

Demonstration

NASA Partnership

- Space Act Agreement (SAA): Intelligence modeling and runtime environment for autonomous robotic systems



- Procedural Reasoning System
- No programming
- Require flexibility to add new behavior, etc.



Applications

- Application: Color Configurator
 - Refrigeration cabinets sales reps use the color configurator to change cabinet color schemes on the fly
- Application: Emergency Action Plan Generator
 - 10,000 sports productions per year
 - Different venues, different production teams
 - Every production requires new Emergency Action Plan
- Application: Performance Tracker
 - Quality and Sales reps
 - Collect product quality information and market intelligence with Blackberry's
 - Aggregate and analyze data for management reporting