

Aura Visual Studio

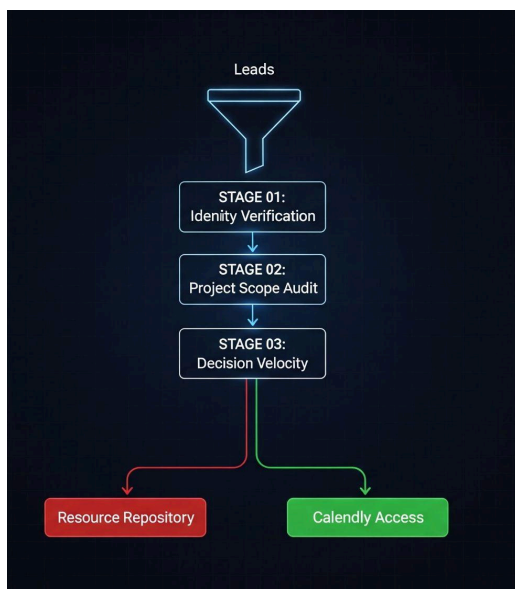
dayshafreeman@auravisualstudio.com
www.auravisualstudio.com

The_Lead_Logic_Manifest // v.01

April 14, 2026

1. Objective: The Quality Filter

The primary function of this logic is to protect your most valuable asset: **Time**. The node must execute a series of invisible checks to verify a lead's "Authority, Budget, Need, and Timeline" (BANT) before they ever see your calendar link.



2. The Logic Architecture

The system operates on a recursive three-stage filter. If a lead fails any single check, the node exits the sales flow and routes them to a lower-tier resource (like a blog or a generic FAQ).

Stage	Action	Target Data Point
01: Initialization	Identity Verification	Business Name, URL, LinkedIn Profile
02: Friction Check	Project Scope Audit	Budget Bracket, Technical Requirements
03: Final Gate	Decision Velocity	Implementation Timeline, Priority Status

3. The Qualification Script (The "Branching" Tree)

To build this into your AI agent, use these specific logic branches:

- **Branch Alpha (The Budget Gate):**
 - *Agent Query:* "What is the capital allocation for this phase of infrastructure?"
 - *Logic:* If < \$X, XXX, route to **Resource Repository**. If > \$X, XXX, proceed to **Branch Beta**.
- **Branch Beta (The Problem Fit):**
 - *Agent Query:* "Are we optimizing an existing system or architecting from Node 0?"
 - *Logic:* If "optimizing," pull specific CRM/API variables. If "Node 0," pull architectural goals.
- **Branch Gamma (The Timeline Filter):**
 - *Agent Query:* "Our deployment window for new nodes is [Date]. Does this align with your launch cycle?"
 - *Logic:* If "No," tag as "Long-term Lead" and move to **Email Sequence**. If "Yes," reveal **Calendly Access**.

4. Data Capture & Handshake

Once the node completes the qualification, the data must be formatted for your CRM. Do not store this as "chat history"; store it as a **Lead Object**.

The JSON Blueprint:

JSON

JSON

```
{  
  "lead_status": "Qualified_A_Tier",  
  "qualification_status": "Passed_All_Filters",  
  "identity_verification": {  
    "business_name": "[Insert Business Name]",  
    "company_url": "[Insert URL]",  
    "linkedin_profile": "[Insert LinkedIn Profile]"  
  },  
  "project_scope": {  
    "budget_verified": true,  
    "budget_bracket": "[Insert Budget Bracket Value]",  
    "architecture_type": "Custom_Node_Deployment",  
    "technical_requirements": "[Insert Technical Requirements]"  
  }  
}
```

```
"integration_points": ["Salesforce", "Stripe", "Custom_API"]
},
"timeline_and_priority": {
  "deployment_urgency": "Immediate",
  "implementation_timeline": "[Insert Target Date]",
  "priority_status": "High"
}
}
```

5. The Routing Fail-Safe

Never leave a lead in a dead-end. If they are disqualified by the logic:

1. **Tag them** in your system (e.g., "Under_Budget").
2. **Redirect them** to a secondary node (e.g., "Download our Strategy_Node PDF").
3. **Terminate the session** to save your API credits.

This ensures your system remains **operationally profitable** while still providing value to the public at no cost to your manual attention.