Eugene Owusu Project - Developer Advisory Meeting

Date: Tonight **Duration:** 45 minutes **Purpose:** Get technical validation and advice on where to start

---

Quick Context (2 minutes)

Who is Eugene?

- Commercial Real Estate Debt Advisor in Berlin (AMA Capital)
- Handles acquisition, development, and refinancing deals
- Currently: 3 deals at a time, wants to scale to 10-15
- Quote: "Dude, my whole business is like 90-95% documents"

The Problem:

- Spends 5-10 hours PER DEAL just in the qualification stage
- Manual document classification (opens 18+ PDFs to identify them)
- Manual ChatGPT uploads and analysis
- Manual follow-up email drafting
- **This workflow doesn't scale**

The Goal: Automate the document-heavy qualification process so Eugene can handle 3-5x more deals

---

Eugene's Current Manual Workflow

(See Diagram 1 attached)

```
Lead comes in via email/call
         ↓
Eugene sends email: "Please send these documents..."
         ↓
Client emails 18+ unlabeled PDFs
         ↓
Eugene downloads ALL files
         ↓
Opens EACH PDF to identify: "Is this the Grundbuch? No, this is BWA..."
         ↓  (1–2 hours)
Manually checks against mental checklist
         ↓
Uploads all docs to ChatGPT (one by one)
         ↓  (30–45 minutes)
Waits for analysis (SWOT, red flags, financial ratios)
         ↓
Manually drafts follow–up email for missing docs
         ↓  (15–30 minutes)
Sends to client
         ↓
REPEAT until complete
```

Total Time: 5-10 hours per deal, mostly manual document handling

---

What We're Proposing

Instead of automating everything at once, we want to break it down into **13 micro-steps** over 3/6/12 months, starting with the smallest possible automation.

Month 1-3 (Foundation)

1. Email Templates (1-2hr) - 3 profile-based templates in Pipedrive
2. Single Doc Classification (2-3hr) - AI identifies JUST Grundbuch documents
3. Pipedrive → Motion sync (3-4hr) - Basic one-way integration
4. Top 5 Docs (2-3hr) - Expand AI to 5 document types
5. Completeness Checker (2-3hr) - Track which docs are missing
6. Custom Field Sync (2-3hr) - Sync more data between systems

Total: 15-18 hours | Value: Save 1-2 hours per deal

Month 4-6 (Expansion)

7. Auto-Reply Drafts (2-3hr) - AI drafts "missing documents" email
8. Deal Analysis (4-6hr) - Full ChatGPT-style analysis automation
9. Follow-up Questions (2-3hr) - AI generates follow-up questions

Total: +8-12 hours | Value: Save 3-4 hours per deal

Month 9-12 (Advanced)

10. Motion → Pipedrive Reverse Sync (8-12hr) - True bidirectional sync
11. Credit Check Automation (12-16hr) - Auto-extract financial ratios

Total: +20-28 hours | Value: Save 4-5 hours per deal

(See Diagram 2: Roadmap Timeline)

The Big Technical Question: Make.com vs n8n vs Custom Code?

This is where I need your advice most.

Option A: Make.com

- Visual workflow builder (like Zapier)
- All integrations validated (Gmail, Claude API, Google Drive, Pipedrive, Motion)
- **Pros:** Fastest to build, easiest to maintain, non-technical friendly
- **Cons:** $11/mo after free tier, less flexible for complex logic
- **Build Time:** 15-18 hours for Month 1-3

Option B: n8n (self-hosted)

- Open source workflow automation
- More flexible than Make.com
- **Pros:** Free (just $5-10/mo VPS), full control, community workflows exist
- **Cons:** Requires technical setup, maintenance overhead
- **Build Time:** 18-22 hours for Month 1-3 (includes setup)

Option C: Custom Code

- Node.js or Python + APIs

- **Pros:** Ultimate flexibility, Eugene owns the code
- **Cons:** 2-3x longer build time, ongoing maintenance
- **Build Time:** 40-60 hours for Month 1-3
- **Code Estimate:** 900-1,400 lines

MY QUESTIONS FOR YOU:

1. What would you recommend for someone building this themselves?
2. Should I start with Make.com and migrate later, or go straight to n8n/custom?
3. What are the gotchas I'm not seeing?

Example: Let's Deep Dive ONE Micro-Step

Micro-Win 3A: Single Document Classification (Grundbuch only)

What it does:

- Watches Eugene's Gmail
- When email arrives with PDF attachment
- AI (Claude API) asks: "Is this a Grundbuchauszug (land registry)?"
- If YES: Save to Google Drive → "Grundbuch" folder
- Email Eugene: "✅ Grundbuch received from [client]"

The Technical Stack:

```
Gmail API (watch emails with attachments)
    ↓
Extract PDF
    ↓
Claude API (vision model, reads German PDFs)
    ↓
If Grundbuch detected
    ↓
Google Drive API (upload to organized folder)
    ↓
Gmail API (send notification)
```

Validation:

- ✅ Gmail API supports attachment extraction
- ✅ Claude 3.5 Sonnet has PDF vision (can read documents)
- ✅ Google Drive API supports folder creation/upload
- ✅ Cost: ~$0.01-0.02 per document

Build Time: 2-3 hours in Make.com

MY QUESTIONS FOR YOU:

1. Are there technical gotchas here I'm missing?
2. Is Claude the right choice for German document classification, or should I test alternatives?
3. How would you handle failures (e.g., API down, misclassification)?
4. Should I build a fallback/review queue?

(See Diagram 3: Technical Architecture)

The Critical Challenges I've Identified

Challenge 1: Motion API Has No Webhooks

- **Problem:** Can't get real-time updates when Eugene completes tasks in Motion
- **Workaround:** Poll Motion API every 15 minutes
- **Impact:** 15-minute delay in syncing back to Pipedrive
- **Question:** Is this acceptable, or should I rethink the Motion integration entirely?

Challenge 2: German Document Accuracy

- **Risk:** AI misclassifying German CRE documents could cause real problems
- **Mitigation:** Start with 1 doc type, manually validate 20-30 samples before expanding
- **Question:** What accuracy threshold would you consider acceptable? 95%? 98%?

Challenge 3: Email → Deal Matching

- **Problem:** Eugene has 10-15 deals active. Client emails 18 documents. Which deal do they belong to?
- **Options:**
  - Parse email subject line for deal name?
  - Require Eugene to manually assign when email arrives?
  - Use sender email to lookup deal in Pipedrive?
- **Question:** What would you do here?

Challenge 4: Cost Management

- **Concern:** Claude API costs ~$1-3 per deal × 10-20 deals/mo = $20-60/mo
- **Question:** Should I build usage monitoring from day 1, or wait until it's a problem?

Where Should I Start? (This is the KEY question)

I've broken everything down into 13 micro-steps, but I'm not sure if:

1. I'm breaking things down too much (should I combine some steps?)
2. I'm starting with the right thing (is email templates really the easiest?)
3. The order makes sense (should I do doc classification BEFORE Pipedrive → Motion?)

HERE'S WHAT I THINK:

- **Start:** Micro-Win 1A (Email Templates) - No code, immediate value, 1-2 hours
- **Then:** Micro-Win 3A (Single Doc Classification) - Proves AI concept, 2-3 hours
- **Then:** Expand from there based on feedback

MY QUESTIONS FOR YOU:

1. Does this order make sense, or would you start differently?
2. Are there dependencies I'm missing?
3. What would YOU build first if this was your project?
4. Am I overcomplicating this?

Testing & Validation Strategy

How I'm thinking about this:

- Build each micro-step in isolation
- Test with Eugene on 1-2 live deals
- Validate accuracy before expanding
- Get feedback before moving to next step

MY QUESTIONS FOR YOU:

1. Should I build a sandbox environment, or test with live deals?
2. How do I test AI accuracy without a large dataset?
3. What failure modes should I plan for from day 1?
4. How much monitoring/logging should I build upfront?

Deployment & Maintenance

Current Thinking:

- If Make.com: Just runs in the cloud, minimal maintenance
- If n8n: Deploy to DigitalOcean/Hetzner VPS ($5-10/mo)
- If custom code: Railway or Render ($5-10/mo)

MY QUESTIONS FOR YOU:

1. What's the simplest deployment for someone building this themselves?
2. Should I use serverless (Cloud Functions) or a VPS?
3. How do I handle secrets (API keys) securely?
4. What happens when something breaks and Eugene needs support?

What I Need From This Meeting

Validation:

- Am I approaching this the right way?
- Are there obvious problems you see that I'm missing?

Guidance:

- Where should I actually start?
- What would you do differently?
- What's the biggest risk here?

Technical Advice:

- Make.com vs n8n vs custom code for my skill level?
- How to test AI accuracy without massive datasets?
- How to handle the Motion webhook limitation?

Methodology:

- How do I prevent scope creep on micro-steps?
- When do I say "good enough" and move on?
- How much documentation/testing should I build upfront?

What Success Looks Like

After this meeting, I want to:

1. Know confidently where to start (which micro-win to build first)
2. Understand the technical approach (platform choice validated)
3. Have a clear sense of risks and how to mitigate them
4. Feel confident I'm not missing something obvious

For Eugene (after we build):

- Month 3: Handling 5-6 deals (vs 3 now) - 2x capacity
- Month 6: Handling 8-10 deals - 3x capacity
- Month 12: Handling 12-15 deals - 4-5x capacity
- **Revenue impact:** Potential 3-5x increase

Final Question for You

If you were starting this project tomorrow, what would you do first and why?

I really appreciate you taking the time to review this and provide guidance. Looking forward to your perspective!