Product Requirement Document

Chosen problem - Agentic chaser for data state management, Follow-ups and Compliant Investments

Solution overview : Jarvis of AdvisoryAI providing data points for investment recommendation to Dr. Strange's timestone i.e Agentic Chaser that look into the future to see which investment products will become "unicorns" and which will fail, eliminating investment risk and analyze years of market volatility in seconds to time exits and entries perfectly, avoiding market downturns.

A State Intelligence Engine that continuously monitors data completeness, automatically chases missing information via SMS/WhatsApp/Email, updates multiple systems simultaneously, and generates compliant investment recommendations—all while maintaining a complete audit trail for FCA compliance.

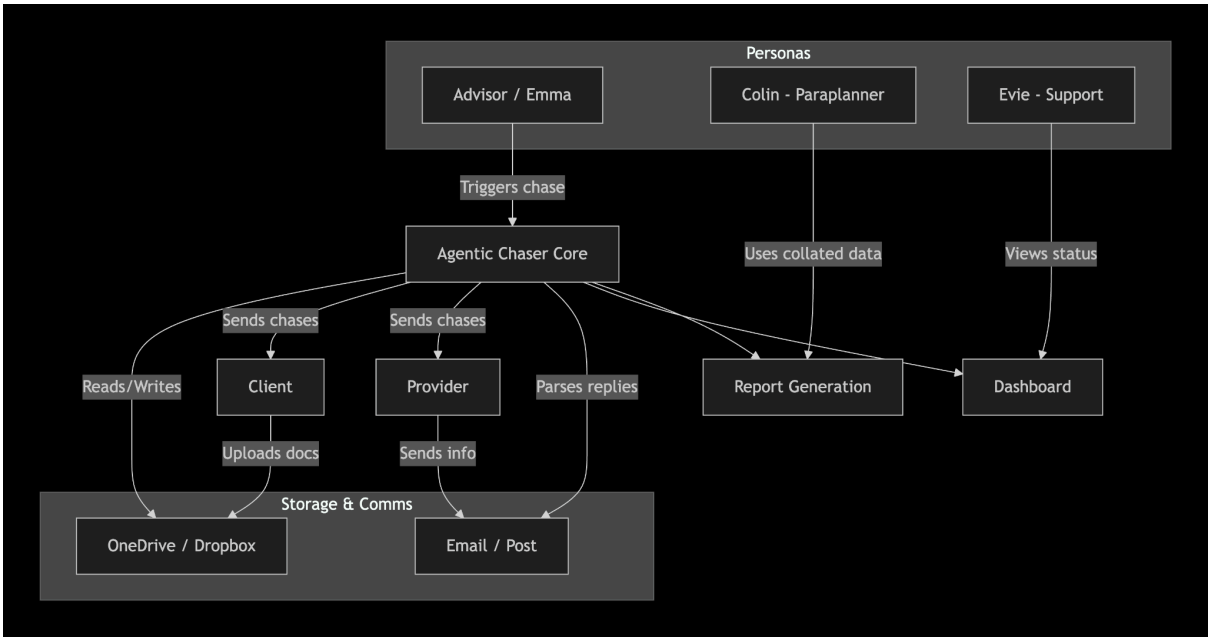
Manual Process Pain Points:

1. State Tracking: Advisor manually tracks 15-20 LOAs across multiple providers
2. Communication Chaos: Chasing via email, phone, SMS with no unified view
3. Data Silos: Information scattered across 6+ systems
4. Compliance Risk: Manual processes create FCA/MiFID II compliance gaps
5. Time Waste: 15+ hours/week per advisor on chasing alone

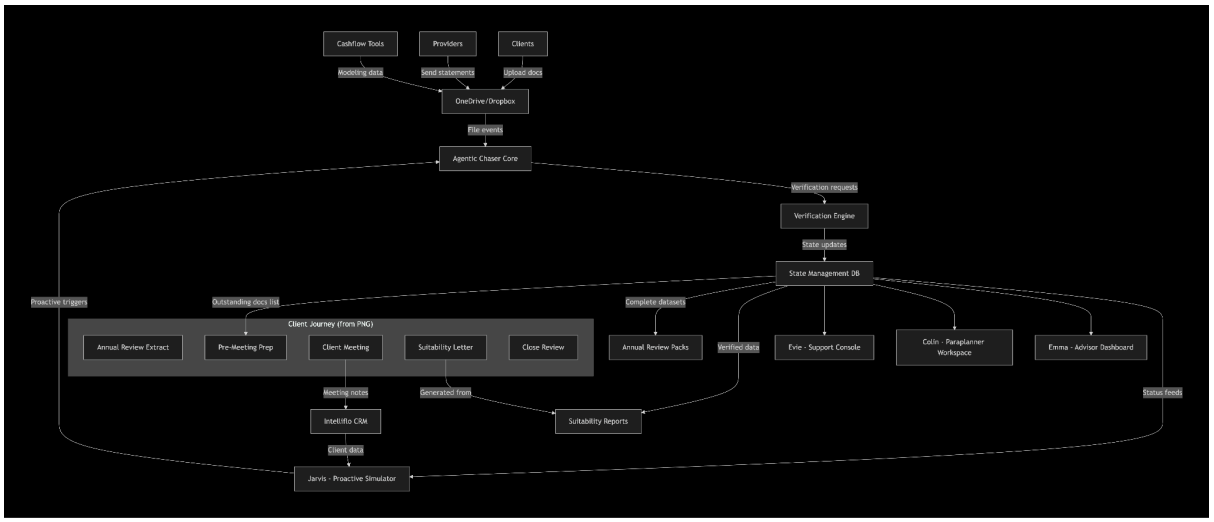
Communication Channel Strategy

Channel	Use Case	Timing	Personalization
SMS	Urgent reminders, quick updates	Business hours, 9am-6pm	Client name, specific item
WhatsApp	Document sharing, informal updates	9am-8pm (client consent)	Rich media, PDF previews
Email	Formal requests, detailed info	9am-5pm	Template with branding
Phone	Complex issues, sensitive topics	Scheduled slots	Advisor voice, rapport

High-level Agentic Chaser system workflow



Dataflow-diagram for Agentic Chaser:



WITH EVIE: Real-time Meeting Intelligence

1. Client Meeting Starts
2. Evie records audio automatically
3. Real-time transcription & analysis
4. Immediate identification of:
   - Document mentions
   - Action items

- Client concerns
- Compliance flags

5. Agentic Chaser triggered in real-time
6. Automated chase begins immediately
7. All systems updated before meeting ends

WITHOUT EVIE: Manual Input Workflow - (Additional delays and increased cost)

1. Client Meeting Occurs
2. Advisor takes manual notes (paper/digital)
3. Meeting ends
4. Advisor uploads notes to system (delayed)
5. Batch Processing:
   - Daily/Weekly processing queue
   - OCR for handwritten notes
   - NLP analysis of notes
6. Missing information identified (24-48 hour delay)
7. Advisor review required for chase approval
8. Manual chase initiation or automated with oversight
9. Systems updated with delay

Client Meeting → Manual Note Entry (Advisor) → Note Upload to Intelliflo →
Scheduled Batch Processing → Jarvis OCR & NLP Analysis →
Missing Data Flagged → Daily Digest Report →
Manual Advisor Review → Selective Chase Approval → Agentic Chaser Execution

Quantitative Success Metrics:

Metric	Current Baseline	Target	Measurement Method
Time per LOA case	4 hours manual	30 minutes	Time tracking in dashboard
LOA completion time	3-6 months	< 30 days	Case timestamp analysis
Document completeness rate	60%	95%	Verification engine logs
Adoption rate (1st month)	N/A	70% of advisors	Login/usage analytics

CSAT (Client)	Industry avg: 3.5/5	4.5/5	Post-case surveys
Advisor time saved	0 hours	15+ hours/week	Pre/post time tracking
Chase Response Rate	40%	85%	Communication analytics
Added Revenue	0 Clients	Based upon time saved no. of clients added	Subscription fees per client

7.2 User Adoption Rate Calculation:

Active User Criteria (weekly):
1. Logs in ≥ 3 times
2. Initiates ≥ 2 chases
3. Reviews dashboard ≥ 5 times
4. Uses auto-collation for ≥ 1 report

```
adoption_rate = (active_users / total_advisors) * 100
```

7.3 CSAT Measurement:

Post-case survey (5-point scale):
1. How satisfied with communication speed?
2. How clear were document requests?
3. How helpful were status updates?
4. Overall satisfaction with process?

```
avg_score = sum(responses) / len(responses)
return avg_score  # 1-5 scale
```

7.4 Time Saved Calculation:

Baseline: Manual process hours per case
Current: Agentic Chaser hours per case

```
time_per_case_saved = baseline_hours - current_hours
total_weekly_saved = time_per_case_saved * number_of_case
```