

HACKFUSION 2026

TEAM IDENTITY

- TEAM NAME – OUTLIERS
- TEAM MEMBERS –
 - DAKSH JAIN
 - VINAY VORA
 - MOKSH JAIN
 - KRISTINA PUJARY
- YOUTUBE LINK – <https://youtu.be/hefvlozel3m>

SOLUTION STRATEGY

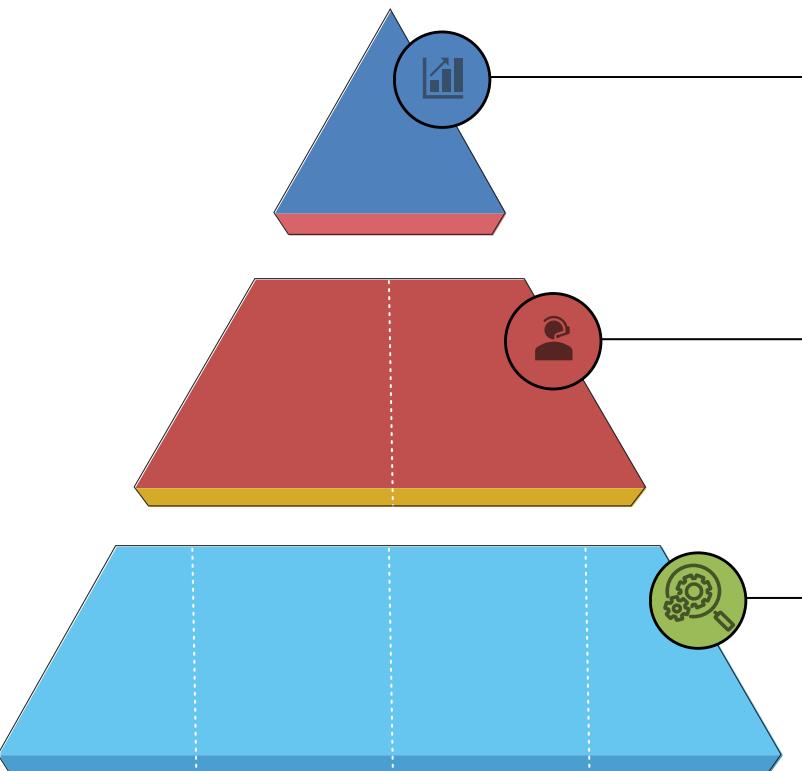


UNDERSTANDING PROBLEM STATEMENT

- **Execution gap** → developers work in tools, but leaders see only delayed status updates.
- **Intelligence gap** → raw signals (commits, tickets, builds) don't become insights.
- **Coordination gap** → teams operate in silos with weak cross-role alignment.

Organizations suffer from Unpredictable delivery, Poor visibility into cost vs output, Weak resource planning, Misaligned teams (engineering vs product vs sales vs finance), Reactive firefighting instead of proactive planning

KEY INSIGHT OR ASSUMPTION

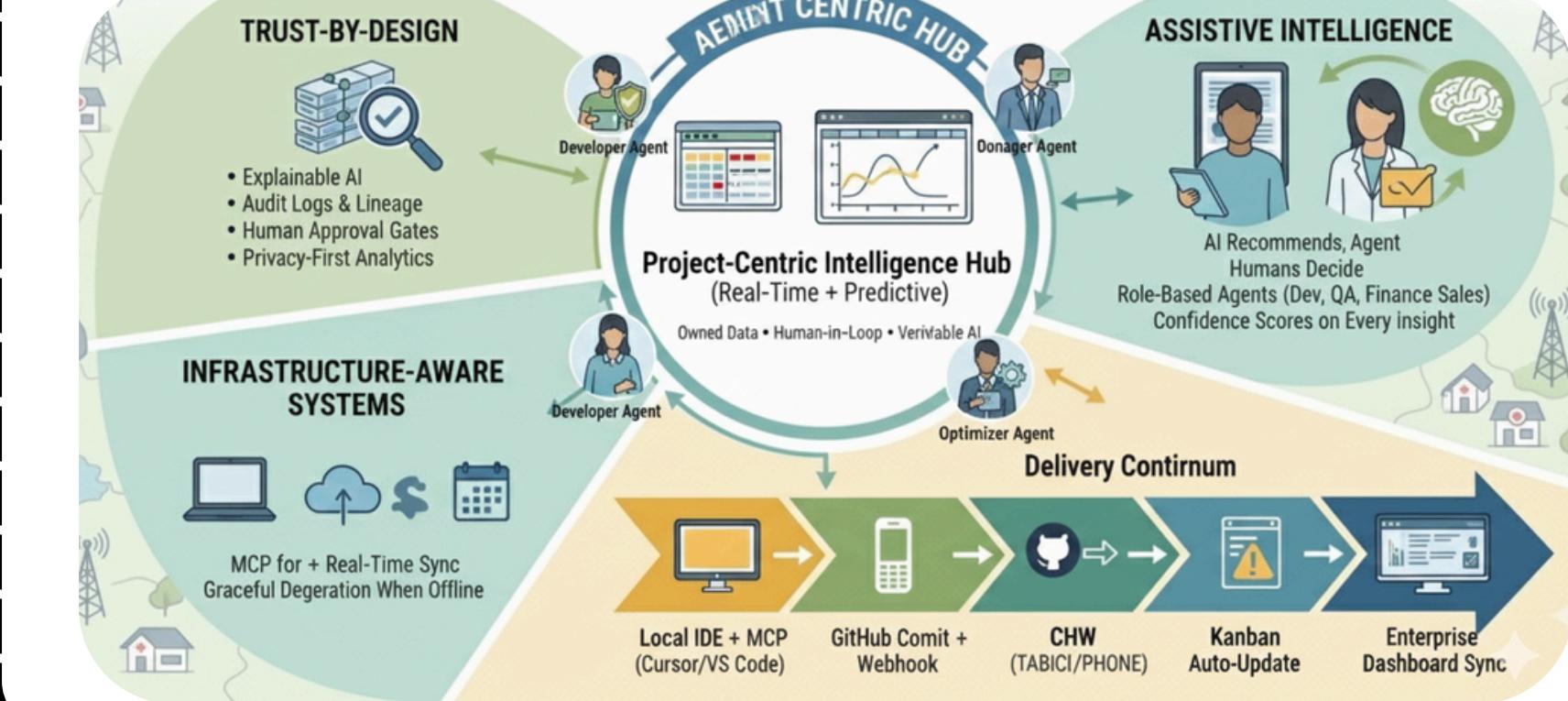


Retrospection alone Is not Business- The biggest value comes from real-time predictive statistics.

AI must assist, not replace humans
-We place emphasis in human-in-the-loop design.

Delivery is a system, not individuals- Delays are usually systemic, not individual performance failures.

APPROACH / PHILOSOPHY



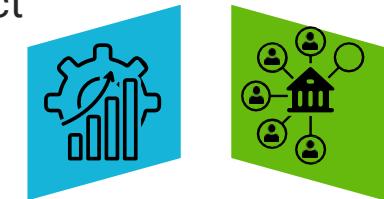
PRIMARY TRADE-OFF CONSIDERED

Trade-off Dimension	Option Rejected	Option Chosen	Rationale
Automation v/s Control	AI autonomously schedules, assigns, and finalizes all project decisions.	Human in the loop for all reviews and approvals.	Ensures trust and responsibility while still delivering efficiency gains.
Deep Analytics v/s Scalability	Deep manual modeling of every single task and dependency.	Monte Carlo + Thomson Sampling	Ensures system is scalable without losing out computationally.
Tools v/s Agent	Only dashboard PM system.	MCP with IDE, GitHub webhooks, and automation.	Proactive intelligent partners over automated tools.

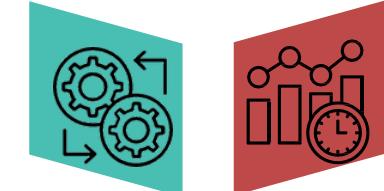
KEY FEATURES

WHAT GAP DOES THIS FEATURE ADDRESS?

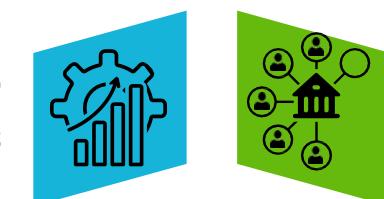
MCP-Enabled IDE Collaboration –
Developers can access tasks, project context, and AI planning directly inside VS Code/Cursor via a unified MCP server.



Intelligent Engineering-to-Business Mapping – Delayed development is automatically converted into delivery risk and cost impact.



Predictive Delivery Intelligence –
Monte Carlo simulation forecasts delays weeks or days before..

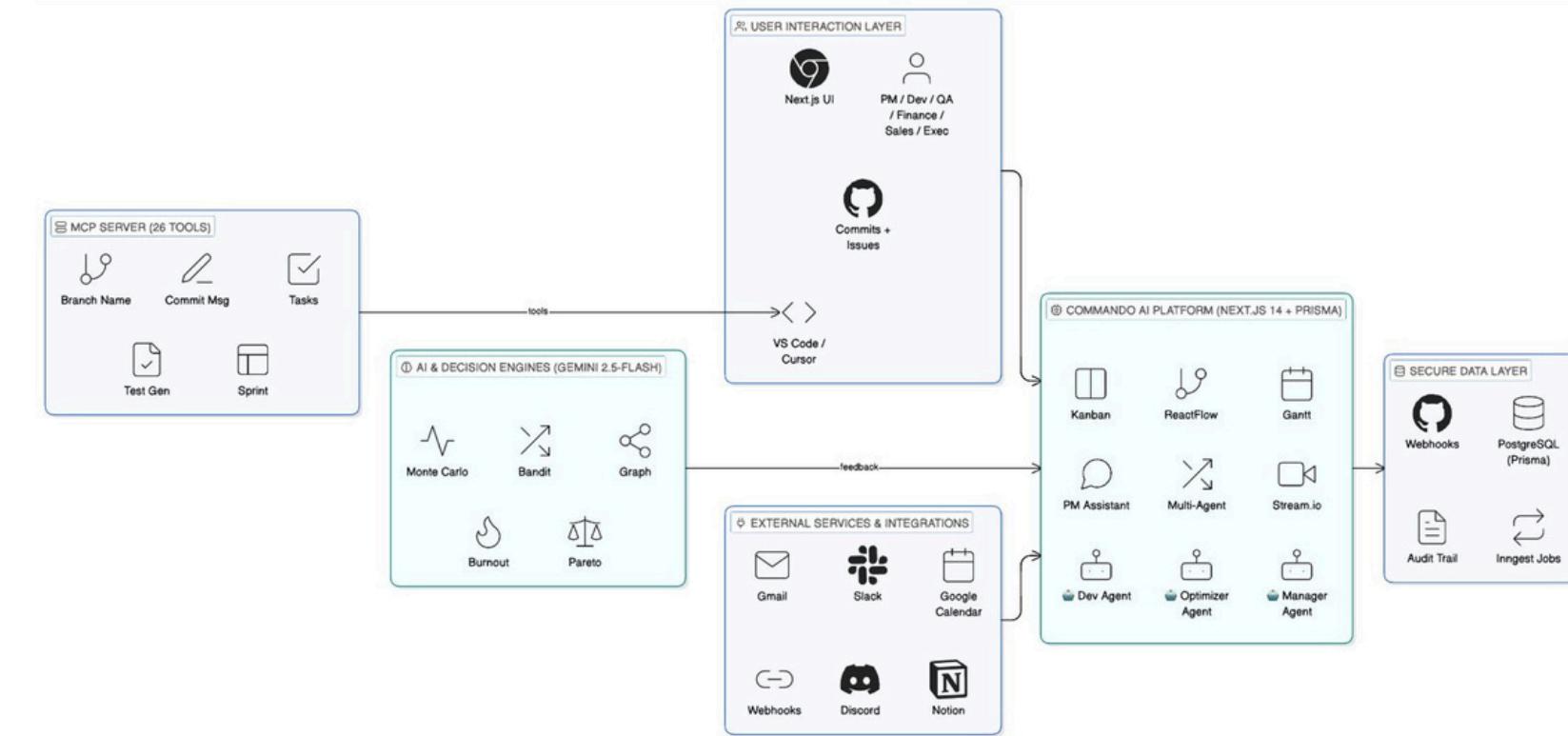


Unified Multi-Role Intelligence – All key users view the same project reality through role-specific dashboards.

AI-Driven Workforce Optimization –
Workload is balanced dynamically using burnout modeling, capacity prediction, and skill-task matching.

Autonomous Status Automation –
GitHub commits, code reviews, and test results automatically update Kanban boards..

HIGH-LEVEL LOGIC OR MECHANISM



OUR USPS

MCP + IDE Integration → we move beyond simple code generation into autonomous, context-aware software engineering.

Predictive Delivery Engine → we use superior Monte Carlo Algorithms than traditional Gantt charts for key delivery statistics

End to End Engineering CRM Platform → we unify sales and engineering to achieve high levels of productivity.

Real World Issue Creation → our active agent ensures issues like managing inventory, optimizing supply chains, or providing personalized customer service are prioritized in real-time.

Thomas Sampling → we dynamically assign the right person to the right task based on historical performance, skill fit, and burnout risk.

WHAT REAL-WORLD IMPROVEMENT DOES THIS CREATE?

Faster Delivery → next to zero delays assisted by automatic workflows ensure client satisfaction

Cost-Aware Delivery Planning → Finance teams track cost per feature and receive early warnings about overruns.



Risk-Based Testing Automation → QA agents prioritize high-risk modules based on commit patterns and historical defects.

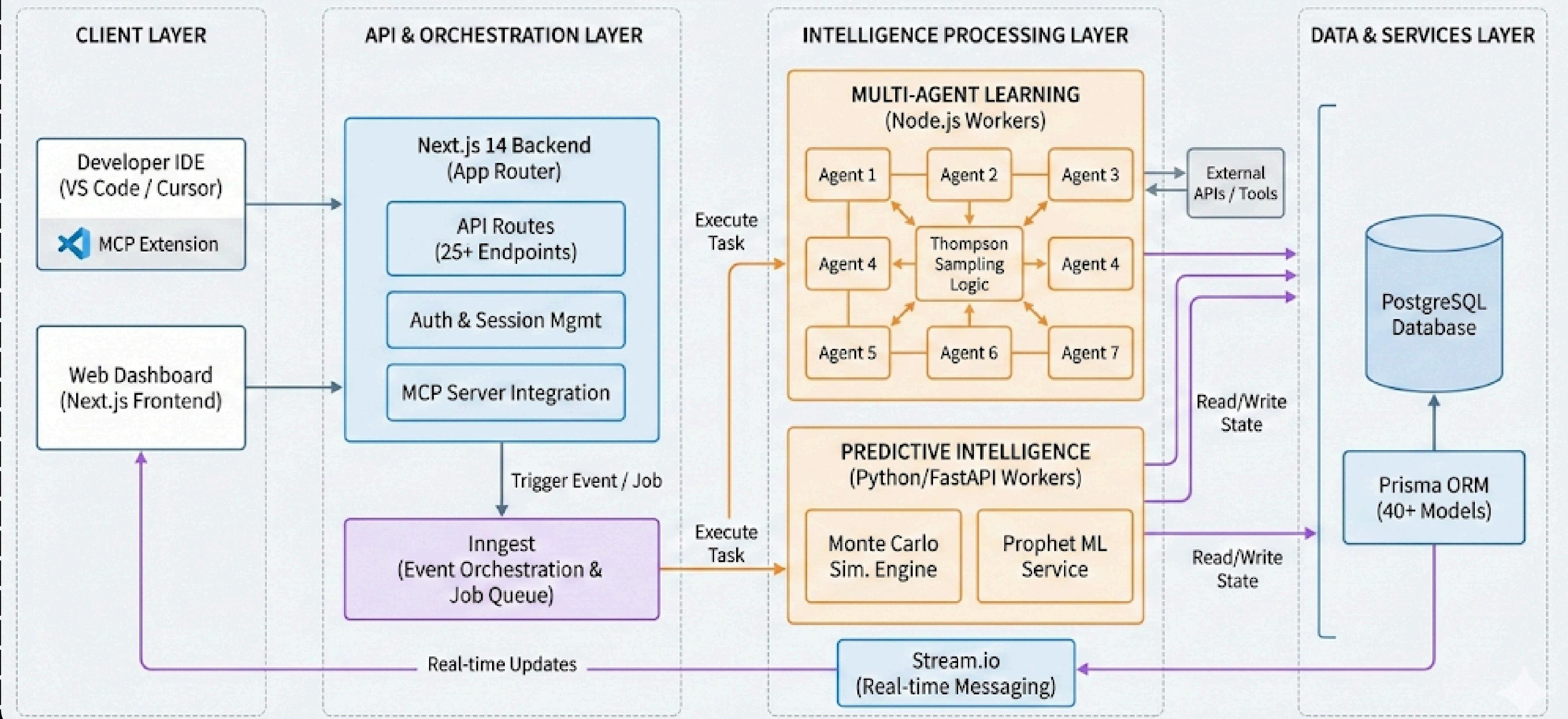
Human-Centered Resource Balancing → Workloads are continuously optimized to prevent overloading critical team members.

Transparent Delivery Promises → Sales teams share AI-calculated confidence scores instead of overpromising timelines.

Single Pane of Truth → Executives make strategy decisions based on real-time delivery health, cost efficiency, and risk exposure.



ARCHITECTURE MAP





PROJECT LINK

<https://github.com/VirusHacks/kaizen>

