

Understanding APIs: How Software Connects



- Orientation: What this training covers and why it matters for your daily work.



- How APIs fit into the modern software landscape: web, mobile, SaaS, and AI.



- Learning goals: Recognize API concepts and speak confidently in product/dev conversations.



- Know when to ask deeper questions and integrate API thinking into your role.



APIs Are Everywhere: The Hidden Infrastructure



Non-engineers encounter APIs in payment, login, and partner integration discussions



Weather apps, social media bots, and “Sign in with Google” all run on APIs



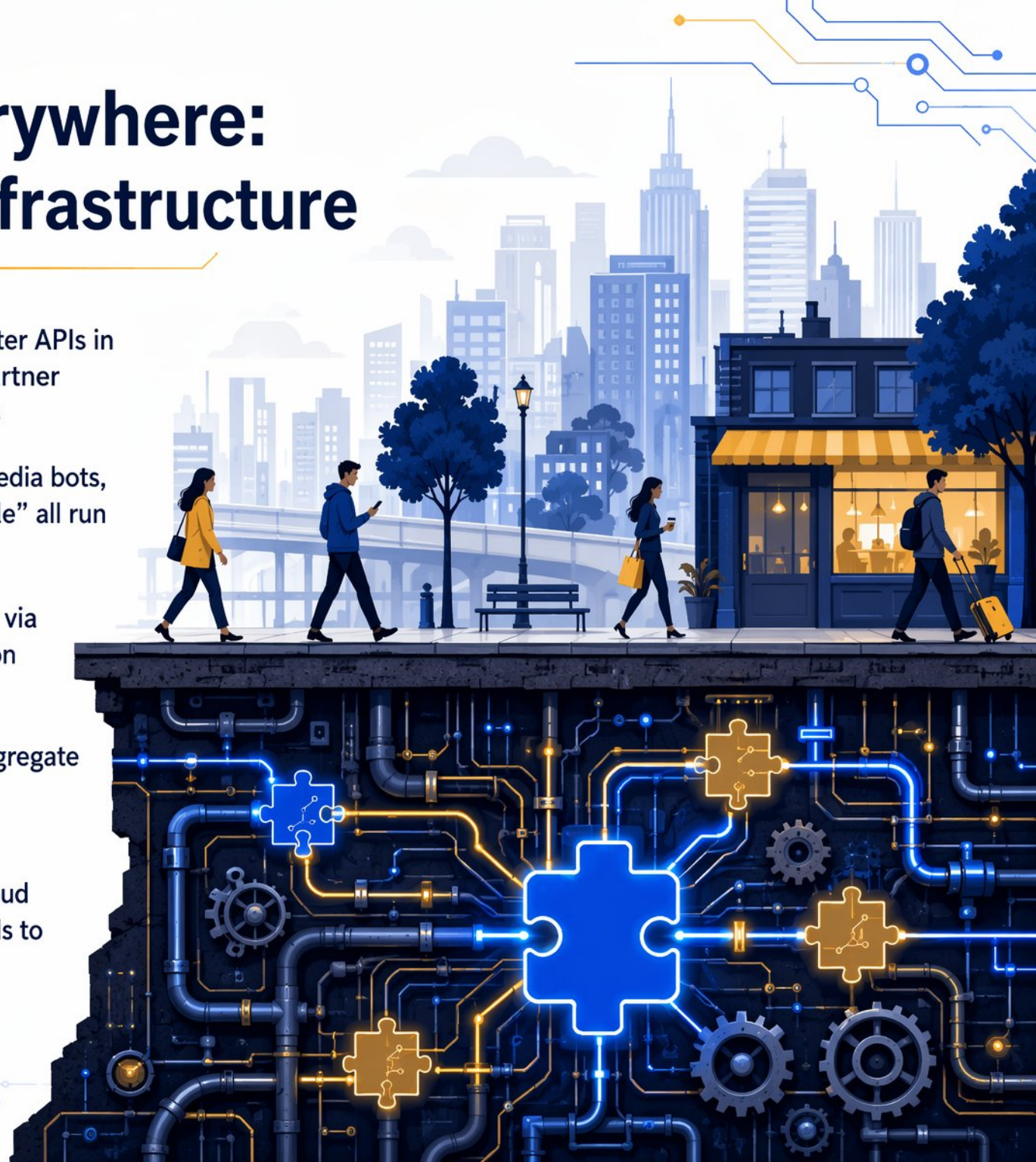
E-commerce checkout via PayPal or Stripe relies on API calls



Travel booking sites aggregate flights and hotels using provider APIs



Banks, logistics, and cloud services use internal APIs to connect systems



What Is an API, Really?



- **Restaurant model:** you (client) order, kitchen (server) prepares, waiter (API) brings the result



- **Formal definition:** a contract that lets two software systems talk to each other



- **Core cycle:** request → processing → response



- **What it's NOT:** a user interface, a database, or a full application



The API Conversation: Request and Response

- Request components: endpoint URL, method (GET/POST/PUT/DELETE), and parameters
- Response: status codes signal success, client error, or server error at a glance
- Walkthrough: requesting a product price — the full request-response lifecycle



Decoding Status Codes:

What the Numbers Actually Mean

- Five classes: 1xx (info), 2xx (success), 3xx (redirect), 4xx (client error), 5xx (server error)
- Restaurant analogy: 200 = order served, 404 = not on menu, 500 = kitchen on fire
- Common codes: 200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error, 503 Service Unavailable
- Golden rule: 4xx errors mean the customer's request is wrong; 5xx errors mean the kitchen messed up



What Data Looks Like: JSON and Authentication Basics



- JSON: Human-readable data format using key-value pairs



- API keys and tokens act like digital ID badges



- Authentication verifies identity; authorization grants permissions



- Example: {"name": "Adventure Mug", "price": 14.99}



```
{  
  "name": "Adventure Mug",  
  "price": 14.99  
}
```



Types of APIs You'll Encounter



- **REST APIs:** 83% of public APIs, resource-based URLs, simple CRUD
- **Webhooks:** push-based, notify you when events happen, no polling
- **GraphQL:** request exactly the data you need, reduce payload by 80%
- **Hybrid reality:** 67% of large orgs use both REST and GraphQL

Reading API Documentation



- Anatomy of a reference page: base URL, endpoint, parameters, and example responses



- The OpenAPI standard: one YAML file describes the entire API contract



- Authentication instructions: where to find API keys, tokens, and scopes



- Practice: scan a doc page to answer a real product question fast



Testing and Troubleshooting Without Code



Open browser DevTools (Network tab) to see live API calls and responses



Use Postman as a GUI client to send requests and inspect results without coding



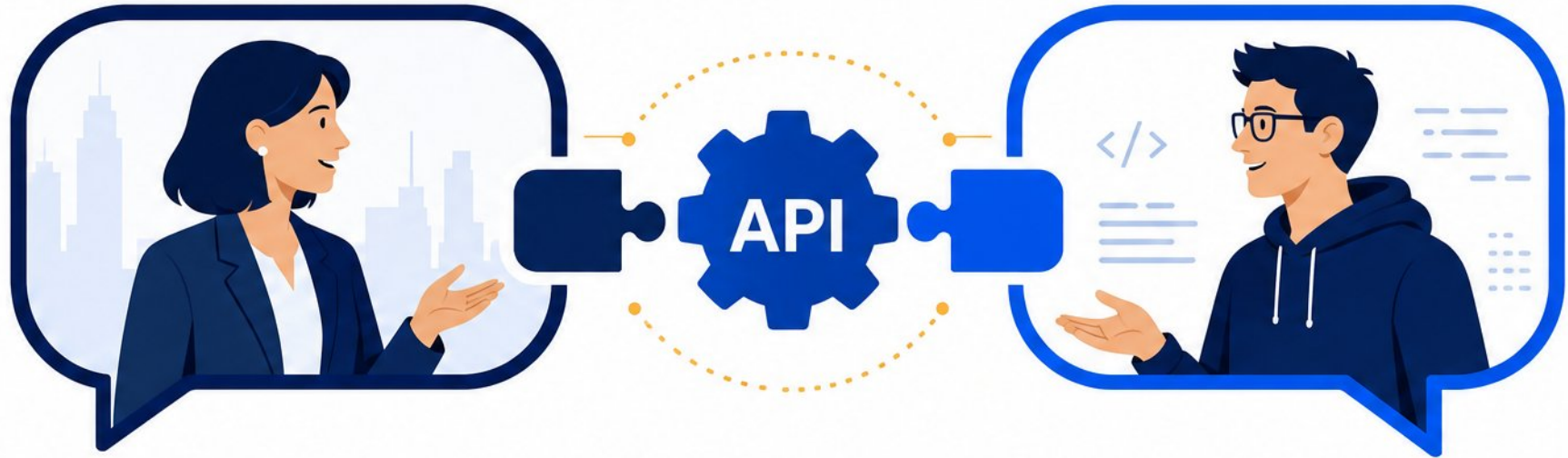
4xx errors = client mistake (bad request, missing auth); 5xx = server failure



When escalating, include the endpoint, method, status code, and response body



APIs in Your Product and Business Conversations



- Partner integration is often really an API integration



- An API exposes data and actions, not the full product



- Ask about rate limits, data formats, versioning, and error handling



- Match business needs to what the API can actually deliver



Security and Trust: What Everyone Should Know



- Exposed API keys can leak data, incur costs, and violate compliance



- Public endpoints without authentication invite service abuse and scraping



- Treat keys like passwords: never share, commit, or hard-code them

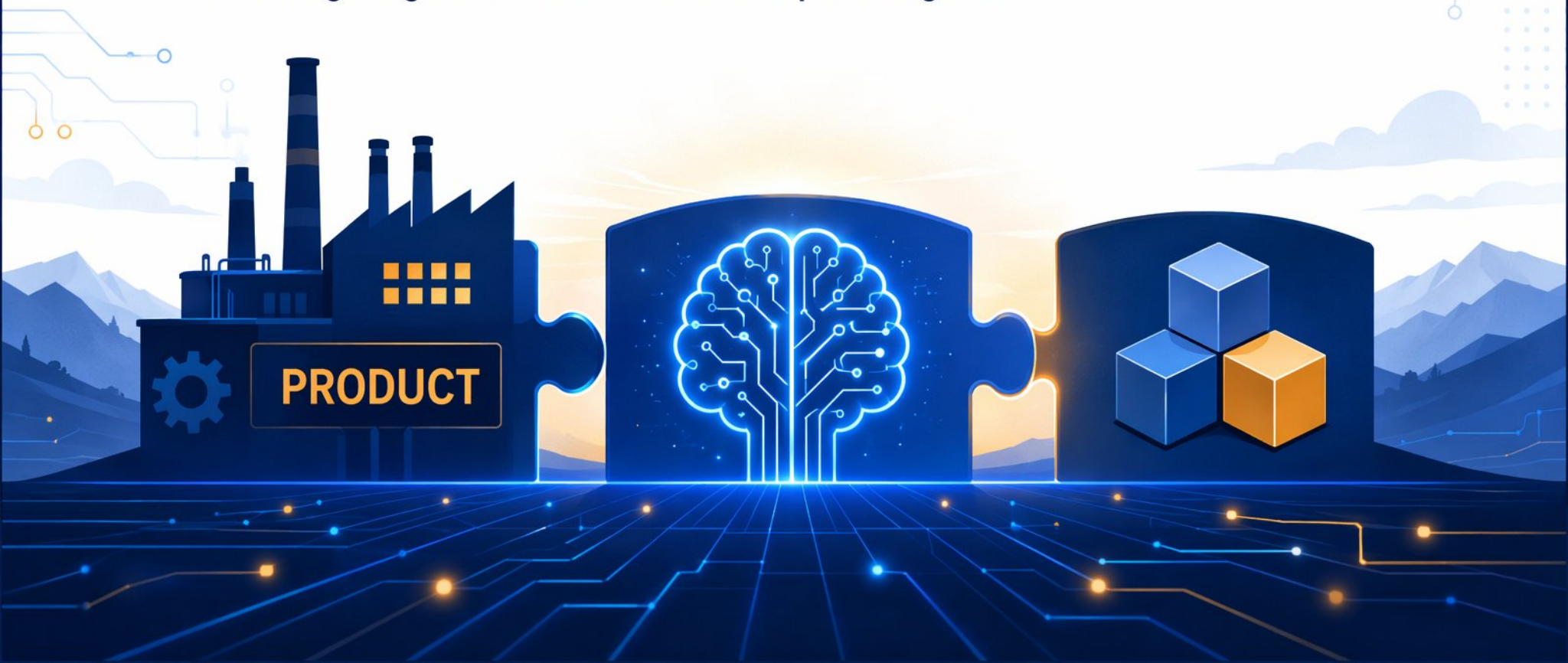


- Use environment variables, vaults, and least-privilege access for all keys



The API Landscape and Future Trends

- - API-first: Treat APIs as products with revenue, not plumbing
- - AI agents consume 8% of API calls—up from 2% a year ago
- - Consumption-based pricing now at 42%, hybrid models dominate
- - Low-code/no-code tools abstract API complexity for all roles
- - 67% of large orgs use both REST and GraphQL together



The AI-API Convergence: A Closer Look



- **Agentic AI as API consumer:** 8% of calls now agentic, up from 2% a year ago



- **APIs redesigned for AI:** Model Context Protocol (MCP) and machine-readable contracts



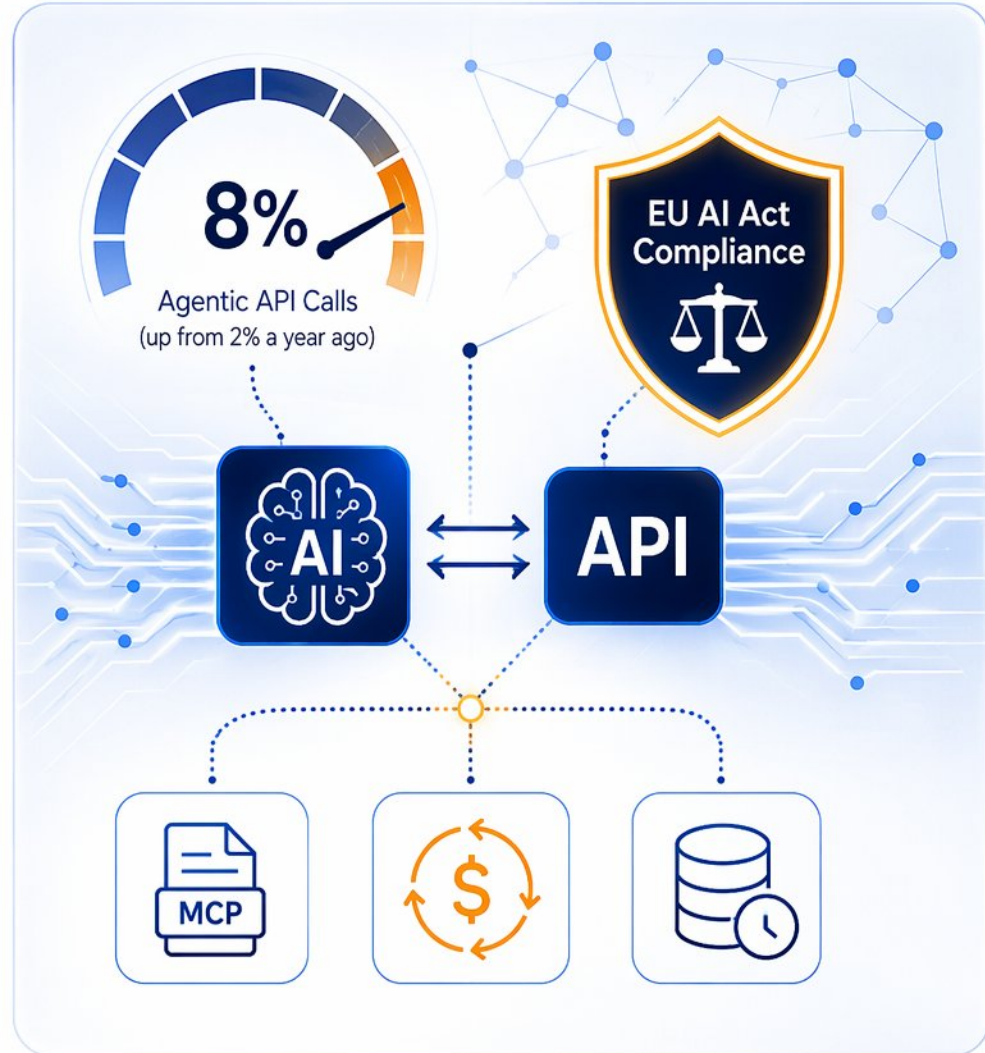
- **New pricing models:** consumption-based, outcome-based, and hybrid billing



- **Compliance shift:** EU AI Act obligations now affect every API integrator



- **Memory and context:** 2M-token windows and cross-session state as API features





The Restaurant Refresher: Key Concepts Review



- The restaurant model: client orders, kitchen prepares, waiter delivers



- Request, response, and five status code families (1xx–5xx)



- REST (83% of public APIs), webhooks (push), GraphQL (ask exactly)



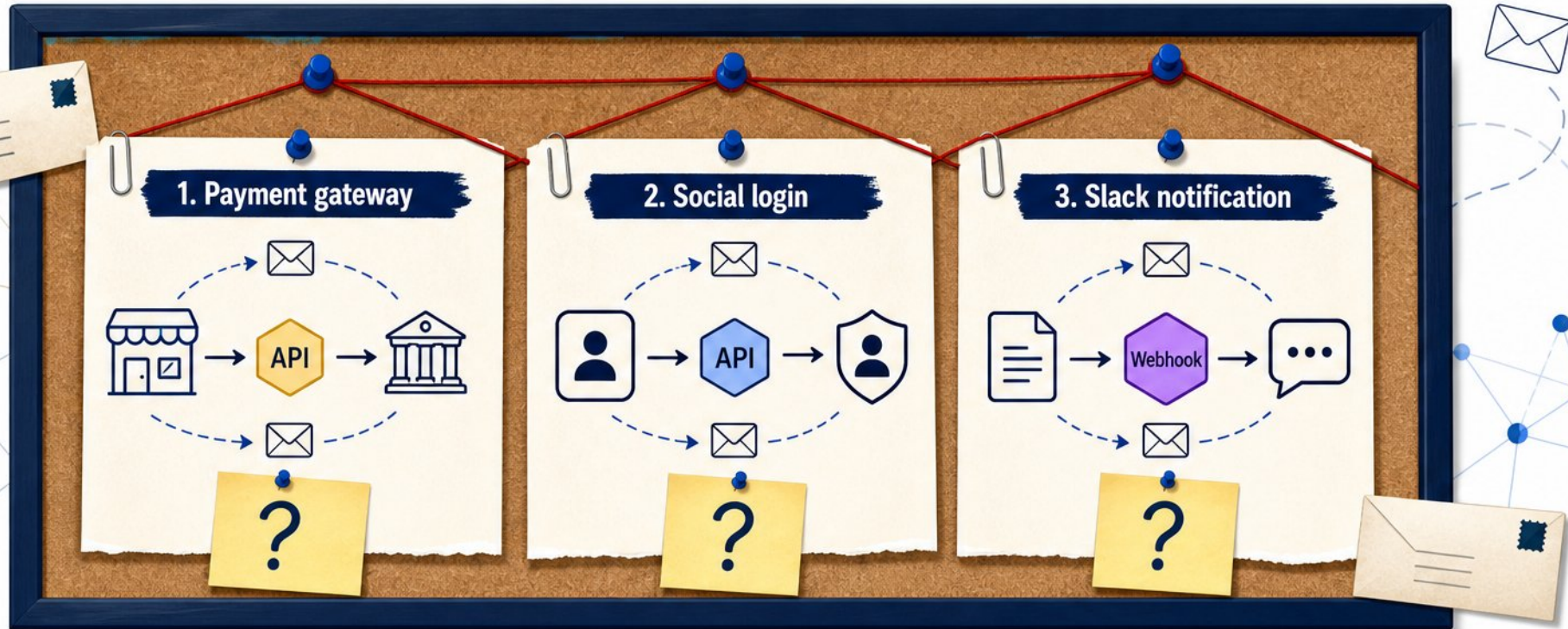
- Authentication: API keys as ID badges; JSON: human-readable data format



- API docs anatomy: base URL, endpoints, parameters, example responses



Practical Scenarios: Spot the API at Work



- Payment gateway: API connects store to processor, handles auth & confirmation



- Social login: OAuth API verifies identity without sharing credentials



- Slack notification: Webhook triggers message when lead is created



- Group task: Identify API types, endpoints, and potential issues per scenario

Where to Go from Here

- - Find one API you use at work and explore its documentation
- - Use Postman's Agent Mode to generate tests from real responses
- - Import an OpenAPI spec into Swagger Editor to see live docs
- - When asking devs for help, share the endpoint, method, and error code



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/623fdd7f/public