# Intelligent Prior Authorization System (IPAS) – Full System Design

## System Overview

The Intelligent Prior Authorization System (IPAS) is an **agentic AI platform** for medical insurance that automates prior authorization decisions for treatments (focusing here on hospitalizations) while incorporating human oversight. The goal is to dramatically reduce the time and effort spent on prior auth, which currently averages ~39 requests per physician per week and 13 hours of staff time weekly[[1]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=in%20late%202024%20found%20that%2C,completing%20those%20requests%20each%20week). By leveraging a GPT-4.1-based AI agent (via Azure OpenAI) and integrating with internal and external data, IPAS can turn a process that often takes **days or weeks** into one of mere **minutes**[[2]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L204%20The%20goal,into%20mere%20minutes%2C%20Rao%20noted), expediting patient care. Crucially, IPAS is designed with transparency and safeguards to avoid the pitfalls of “unregulated AI” – a **human-in-the-loop** review is employed for complex cases so that automated decisions do not override good medical judgment[[3]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Health%20insurers%E2%80%99%20use%20of%20AI,judgment%20and%20exacerbate%20patient%20harm). In short, IPAS aims to streamline prior authorizations (PAs) with intelligence and speed, while **ensuring trust, compliance, and auditability**.

## Data Sources and Integrations

A core strength of IPAS is its ability to pull information from **multiple data sources**, both within the insurer and from external healthcare systems, to inform its decisions. Key data sources include:

* **Electronic Health Records (EHR) Systems:** IPAS connects to common hospital EHRs (via standards like HL7 FHIR APIs) to retrieve patient medical data – diagnoses, lab results, physician notes, medications, etc. This provides clinical context for each request (e.g. severity of illness, co-morbid conditions). We assume connectors to major EHRs (Epic, Cerner, etc.) are available, enabling real-time data access. For example, if a prior auth is for pneumonia hospitalization, the agent can fetch the patient’s oxygen levels and X-ray reports from the hospital’s EHR to assess medical necessity.
* **Insurance Policy & Eligibility Data:** Internal insurance databases provide the patient’s plan details, coverage limits, and eligibility information. This ensures the agent knows whether the requested service is covered by the policy and any required criteria (e.g. needing pre-authorization for certain procedures). The agent can verify member eligibility, plan type, in-network provider status, and any utilization management rules specific to the plan.
* **Claims and Utilization History:** IPAS can look at the patient’s past claims or prior auth history. For instance, if the patient had a similar hospitalization recently or has exhausted certain benefits, the agent factors that in. Patterns in historical data (such as frequent readmissions) might trigger closer review. This data resides in the insurer’s claims management system and is accessible to the agent via secure APIs.
* **Medical Policy Guidelines and Clinical Criteria:** The insurer’s medical policy library and industry clinical guidelines (e.g. MCG/Milliman or InterQual criteria for hospital stays) are used as knowledge bases. IPAS employs Retrieval-Augmented Generation – it can search a repository of **clinical criteria** and prior authorization rules to find relevant guidelines for the requested service. For example, it might retrieve a policy that says “Hospital admission for pneumonia is medically necessary if XYZ criteria are present.” These guidelines, whether internal or external, act as a rulebook for the AI. (In Azure, this could be implemented with Azure Cognitive Search or a vector database of policy documents.)
* **External Databases and Services:** The agent can tap into external sources as needed. This might include verifying provider credentials (e.g. is the provider licensed and in-network?), checking for possible fraud indicators, or referencing medical knowledge bases (such as published literature or drug databases if the request involves certain medications). For instance, if an unusual procedure is requested, the agent could query an external knowledge source to understand it. Integration with external APIs (e.g. a national provider registry or NIH clinical guidelines) expands the agent’s understanding beyond the insurer’s own data.
* **Prior Auth Request Documents:** The prior authorization form itself (often a PDF or image) is a data source. IPAS uses OCR and document parsing (via Azure Cognitive Services, e.g. Form Recognizer) to extract structured information from uploaded **PDFs or scans** of the request. This includes patient info, requested service codes (ICD-10, CPT codes), provider notes on justification, etc. The extracted data is then cross-checked with the other sources above. All documents (clinical notes, attachments) are ingested and categorized automatically, eliminating the need for manual document handling.

All these sources are accessed through **secure connectors** – the system is built on Azure, so it leverages Azure’s integration capabilities (e.g. Azure API Management for internal services, FHIR connectors for EHR, secure storage for documents). The design ensures **seamless integration with existing workflows** like EHRs and claims systems, **without disrupting** them[[4]](https://www.banjohealth.com/#:~:text=Effortless%20Integration). By compiling a holistic picture from internal records and external evidence, IPAS can make well-informed authorization decisions.

## Architecture and Key Components

IPAS is built as a **modular, multi-layered system** on an Azure tech stack. The architecture balances **LLM intelligence, deterministic checks, and orchestration** of workflows. Major components include:

* **LLM Core (Decision Engine):** At the heart is Azure OpenAI GPT-4.1, which serves as the reasoning engine. This large language model is deployed with **appropriate system prompts and guardrails** to carry out medical reasoning, interpret guidelines, and generate decision rationales. All LLM calls are made through Azure’s secure endpoint, ensuring data (which may include PHI) is kept confidential (HIPAA-compliant).
* **Agentic Framework (Neurostack + LangGraph):** We utilize our custom **Neurostack agentic framework** (based on the open-source LangGraph foundation) to orchestrate the LLM and tools. The agent is *agentic*, meaning it can plan multi-step operations autonomously[[5]](https://www.nexastack.ai/blog/agentic-ai-agent-framework#:~:text=Enterprise%20AI%20agents%20are%20autonomous,improvement%20over%20time). LangGraph provides structures for managing the agent’s **memory and tool usage**. The Neurostack framework allows defining sub-agents or tasks as a **graph of actions** (for example: Parse Document -> Retrieve Data -> Verify Criteria -> Make Decision -> Communicate Result). An **Orchestrator Agent** oversees the process, calling specialized sub-agents as needed (e.g., a sub-agent to verify eligibility, another to check clinical criteria). This design is similar to how an AWS agentic AI orchestrator coordinates sub-agents[[6]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=2,agents)[[7]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=11,agent%20performs), but here implemented with Azure services. The framework can run in an Azure Function or Container Instance, ensuring scalability.
* **Memory Modules:** The agent maintains both short-term and long-term memory to improve performance and personalization over time[[5]](https://www.nexastack.ai/blog/agentic-ai-agent-framework#:~:text=Enterprise%20AI%20agents%20are%20autonomous,improvement%20over%20time). Short-term (episodic) memory stores the current conversation and case context, so the agent remembers details as it processes a request. Long-term memory (semantic memory) stores facts and learnings across cases – for example, the agent can accumulate knowledge of how previous similar cases were decided, or update profiles like a particular provider’s typical compliance. This is implemented via a vector store or database (e.g. Azure Cosmos DB or Cognitive Search index) where each case’s key details and outcome are saved. Over time, IPAS “learns” from outcomes (if a denial is frequently appealed and overturned, the agent can adjust its criteria or flag such cases for human review next time – a feedback loop). This continuous learning approach means the AI **constantly adapts to evolving clinical and regulatory requirements**[[8]](https://www.banjohealth.com/#:~:text=Purpose).
* **Tool Integrations (Plugins):** The agent has access to various tools:
* *Document Parser Tool:* Uses OCR to extract text from prior auth PDF/image uploads. Key fields (patient name, ID, diagnosis codes, procedure codes, notes) are identified, possibly using a fine-tuned Form Recognizer model for healthcare forms.
* *Database Query Tool:* Secure connections to internal databases (for eligibility, benefits, claims history). This might be implemented via APIs or direct queries. For example, the agent can call an API getCoverage(patient\_id, service\_code) to see if the service is covered or requires special approval.
* *Knowledge Retrieval Tool:* Performs semantic search over the medical policy library and guidelines. We index documents (policies, clinical criteria, CMS regulations) in a vector store so the agent can retrieve relevant passages. During reasoning, the agent can pull in excerpts (e.g., “Guideline X states hospital stay beyond 5 days requires proof of complication[[9]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=1,survey.pdf)”).
* *EHR Connector:* As mentioned, an API integration (FHIR client) to fetch patient data from the provider’s EHR in real time. The agent might use this to get the patient’s problem list, recent vitals, imaging reports, etc.
* *External API Tools:* For tasks like fraud detection or provider verification, the agent can call external services. E.g., a tool that checks the provider’s NPI and whether they’ve had any sanctions, or a tool to calculate an approximate cost for the requested service (to flag unusually expensive requests).
* *Communication Tool:* An email or messaging service integration to send out the decision. For example, using Azure Communication Services or an SMTP connector, the agent can automatically email the provider and patient with an approval letter or denial notice (including reasons). This tool ensures the final step – communicating the decision – is automated and auditable.
* **Business Rules Engine:** In parallel with the AI’s reasoning, IPAS includes a rule-based engine for straightforward checks. This could be an Azure Function or logic app that enforces hard rules (e.g., coverage exclusions, plan limits). For instance, if a plan *never* covers a certain experimental treatment, the engine can immediately flag that. The AI agent will incorporate these binary outcomes (the agent might be informed “RuleCheck: service not covered per policy” and then it can formulate a denial citing that rule). This hybrid approach (rules + AI) ensures compliance with firm policies and speeds up simple decisions.
* **User Interface:** (Detailed in next section) A web-based interface (likely an Azure Web App or integrated into an adjuster’s dashboard) provides the front-end for interacting with IPAS. It supports chat-based interaction, case viewing, and uploading documents.
* **Case Database & Logging:** All prior auth requests and outcomes are stored in a database (e.g. Azure SQL or Cosmos DB). This serves both as **audit log** (for compliance and quality improvement) and as a source for the agent’s long-term memory. Each case entry might include the extracted data, the agent’s decision rationale, whether human intervention occurred, and final outcome. This repository allows tracking metrics (turnaround time, approval rates, appeal rates) and is useful for regulators or internal QA to review decisions.

Overall, the architecture is cloud-native, leveraging Azure’s scalability and security. The **entire stack is on Azure** (OpenAI for the model, Azure Functions for orchestration, Azure Storage for documents, etc.). The design also aligns with upcoming regulatory mandates for interoperability – by using FHIR APIs and logging detailed decisions, IPAS is compliant with the CMS Prior Authorization Rule (CMS-0057-F) requiring standardized APIs and transparency in decisions[[10]](https://www.precisionaq.com/hubfs/5014803/Modernizing%20PA%20Process.html#:~:text=The%20Catalyst%3A%20CMS)[[11]](https://www.precisionaq.com/hubfs/5014803/Modernizing%20PA%20Process.html#:~:text=The%20Great%20Migration%3A%20From%20X12,to%20FHIR).

## Memory and Continuous Learning

*(This is a special focus given the “agentic” nature of IPAS and use of LangGraph for memory.)*

IPAS’s agent uses memory in several ways to enhance decision-making and personalization:

* **Conversation (Session) Memory:** During a single case review session (which may involve back-and-forth with a human reviewer in chat), the agent retains the dialogue and case facts. It will remember what the user has asked and what it has already explained, to avoid repetition and provide context-aware answers.
* **Long-term Case Memory:** The agent logs salient facts and outcomes from each case into a memory store. Over time, this creates a knowledge base of what happened in various scenarios. For example, the agent can recall that “in 5 past cases of pneumonia requiring >7 day stay, 4 were denied and later 3 of those were overturned on appeal” – this might make it more cautious and suggest human review for similar new cases. Technically, this could use LangGraph’s memory store to save each case as a JSON document, including features like diagnosis, requested length, decision, and appeal result. When a new case comes, a **semantic search** on this memory can retrieve analogous cases to inform the decision (few-shot style guidance or just additional context for the LLM).
* **Profile and Preferences:** The system can maintain profiles – e.g., provider profiles or patient profiles that might affect decisions. If a certain provider has a “gold-card” status (consistently following guidelines, as some insurers offer[[12]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L222%20that%20provides,based%20clinical%20guidelines)), the agent’s logic might automatically fast-track their requests (fewer questions asked). These profiles are updated as the agent observes behavior. Similarly, patient profiles might note chronic conditions or special considerations (like a flagged vulnerability) that the agent should remember across requests.
* **Feedback Loop:** IPAS is designed to learn from its mistakes. All decisions that are later appealed or result in grievances can be fed back to the agent’s memory. For instance, if IPAS denied a request but a human reviewer or appeals board later approved it, the details of that case are used to refine future AI judgments. The agent might store a new rule or adjust the weighting of certain criteria. This **continuous improvement** is crucial for the system to remain accurate and fair as medical knowledge and insurer policies evolve[[8]](https://www.banjohealth.com/#:~:text=Purpose). A governance process would review AI decisions periodically and update the model prompts or memory with corrections (a form of model fine-tuning via feedback, albeit mostly handled at the prompt/memory level given we can’t retrain GPT-4.1 easily).

In summary, memory in IPAS ensures that the AI isn’t a stateless decision tree each time – it builds experience. This aligns with best practices in agentic AI where **memory modules and feedback loops** allow the agent to improve and personalize its behavior over time[[5]](https://www.nexastack.ai/blog/agentic-ai-agent-framework#:~:text=Enterprise%20AI%20agents%20are%20autonomous,improvement%20over%20time). Such memory also helps with compliance and explanation, as the agent can refer back to recorded justifications when answering questions about a decision.

## User Interface: Chat and Document Intake

The primary user interface for IPAS is a **chat-driven application** combined with case management screens. This interface serves internal insurance reviewers (and potentially other stakeholders) and is designed to be intuitive and interactive:

* **Prior Auth Request Intake:** Users (e.g. an insurance intake specialist or the system automatically via an e-portal) start a case by providing the prior authorization request. This can be done by uploading a **PDF or image** of the prior auth form (or via an electronic submission from a provider portal). The UI allows drag-and-drop of the form or fetching it from an email/fax system. Once uploaded, the agent’s document parser tool kicks in to extract the needed information. The UI will show a parsed summary for confirmation – e.g., “*Request from Dr. Smith for Hospital admission of Patient Jane Doe, Diagnosis: Pneumonia, for 3 days starting 10/1/2025*” along with any attached clinical notes.
* **Initial AI Analysis:** As soon as the data is ingested, IPAS performs an **initial analysis autonomously**. The chat interface will display a message (from the AI agent) summarizing the findings and a tentative decision. For example, the agent might say: *“Based on the provided information, the patient meets criteria for inpatient hospitalization due to low oxygen saturation and pneumonia. I recommend* *Approval* *for an initial 3-day stay.”* This message would be accompanied by an explanation or rationale, with references to specific data points or guidelines (the agent can cite the policy it used, or relevant patient data). This gives the human user a quick overview without having to dig through all documents manually.
* **Interactive Q&A:** The human reviewer can now chat with the agent to delve deeper or provide feedback. They might ask follow-up questions in free text, such as “Why exactly 3 days? What if the patient needs longer?” or “What criteria did you check?” The GPT-4.1-based agent will respond in conversational language, pulling in details from guidelines or the patient’s case as needed. For instance, it could answer: *“Three days is the typical initial length for pneumonia per guidelines X. Extension will be approved if follow-up clinical data (e.g. persistent fever or oxygen need) supports it. I checked that the patient’s oxygen was 85% on room air, which satisfies the criterion for acute inpatient care.”* This ability to query the AI builds transparency – the reviewer can understand the basis of the recommendation and even challenge it if something seems off.
* **Simulation Mode (/analyze\_and\_visualize):** A unique feature of IPAS is the **simulation engine** accessible via a special chat command. If the user types **/analyze\_and\_visualize**, the agent will enter a mode where it *simulates the decision process and outcomes* in a more visual or structured way. This might generate a flowchart or step-by-step breakdown in the chat. For example, the agent could display a list of verification steps it conducted:
* Eligibility check – *PASSED* (Patient is active and hospital service is covered)
* Clinical criteria check – *PASSED* (Diagnosis and severity meet guidelines)
* Fraud/red-flag check – *PASSED* (No anomalies detected)
* Confidence score calculation – *High (0.95)*[[13]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=4,These%20criteria%20include)
* Decision outcome – *APPROVE* (auto-communicate to provider)

It could also visualize a **decision tree** or timeline: e.g., a flow diagram showing that if the confidence was below threshold, it would route to human review[[14]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=5,a%20detailed%20justification%20for%20the). The simulation essentially lets the user see *“what the AI is thinking”* and even explore *what-if scenarios*. A reviewer could ask in simulation mode, “Show outcome if oxygen was normal” – and the agent might illustrate that in such scenario, criteria fails and a denial would be recommended. This feature enhances explainability and helps users trust the system’s decisions, by **showing the logic path** rather than just outputting a result.

* **Finalizing the Decision:** After any needed interaction, the user (or the agent automatically, if no human input is needed) will finalize the prior auth decision. This can be done via a simple command or button in the UI (e.g., “Approve”, “Deny”, or “Send for manual review”). In straightforward cases, the agent might finalize on its own and just inform the user (“Approved automatically as criteria 100% satisfied”). In cases where the user was engaged, the user can confirm the AI’s recommendation or override it. The chat history, along with any simulation results, is saved for audit.
* **User Roles:** The interface supports different roles. Internal medical reviewers or nurses will use the chat to handle complex cases. There might also be a view for *external providers* (e.g., a provider could log into a portal to see the status of their request and possibly interact in a limited way to answer questions). However, primarily it’s an internal tool. In complex cases, an internal physician reviewer might jump into the chat to discuss with the AI (almost like consulting a virtual assistant) before making the ultimate decision. The agent is capable of conversing at a clinical level, citing evidence for or against authorization, which can help human reviewers make consistent, well-informed decisions.

Overall, the chat interface brings a **conversational paradigm** to prior auth processing, making it more like collaborating with a knowledgeable assistant than wading through forms. By reading PDFs/images, summarizing them, and allowing natural language Q&A and simulation, IPAS’s UI greatly eases the cognitive load on human staff.

## Decision Workflow and Use Case Scenarios

To illustrate how IPAS works end-to-end, we focus on a **hospitalization prior authorization** scenario. We’ll walk through three related use cases: an initial hospital admission request, and two subsequent extension requests (one approved, one denied). These demonstrate the system’s handling of approvals, partial extensions, denials, and human involvement.

### Case 1: Initial Hospitalization Request (Approved)

**Scenario:** A provider submits a prior auth request for an initial hospital admission. For example, an elderly patient with pneumonia is being admitted to the hospital for treatment. The request is for an **inpatient stay of 3 days** for acute pneumonia care.

**Workflow Steps:**  
1. **Intake:** The hospital’s utilization management team submits the PA request via the portal. The form and accompanying clinical note from the doctor (indicating the patient’s symptoms, vitals, diagnosis) are provided. IPAS ingests this immediately.  
2. **Data Gathering:** The IPAS agent auto-fetches relevant data. It pulls the patient’s insurance plan info (coverage for inpatient care is confirmed, and a note that any stay beyond 5 days requires additional review per policy). It queries the EHR for clinical details: e.g., finds the patient’s oxygen saturation is 85%, chest X-ray shows pneumonia, and the patient has failed outpatient antibiotics – all indicating hospital care is needed. It also retrieves the internal guideline for pneumonia admissions.  
3. **Automated Decision:** GPT-4.1 (via the agent) analyzes the data against criteria. The guideline says inpatient admission is appropriate if oxygen < 90% or other complications present – criteria which this patient meets. The agent also checks there are no *red flags* (no indication of fraud or coverage exclusion). It computes a high confidence score for approval. Since all checks passed, the **agent decides to approve** the request.  
4. **No Human Review Needed:** Because this case was straightforward (clear criteria met, high confidence), IPAS classifies it as **auto-approvable**. It does not require a manual reviewer to intervene. (Internally, the system might mark “Complexity = Low, auto-approved”.) This addresses efficiency: simple cases flow through without waiting on human queues, helping achieve that <10 minute turnaround goal[[15]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=To).  
5. **Communication:** IPAS generates an approval notification. It creates an approval letter detailing the authorization: e.g., “Approved for inpatient hospitalization, up to 3 days, for pneumonia. Reference #: 12345.” The agent includes a brief rationale in the record (e.g., “meets criteria per policy X due to hypoxia”). This letter is automatically emailed to the provider and also made available in their portal.  
6. **Logging:** The case is logged in the database with status “Approved” along with all supporting info (timestamp, the criteria matched, etc.). This can later be reviewed if needed. Also, the agent’s memory stores this outcome (could be used if the patient later requests an extension or if a similar case comes up).

**Outcome:** The patient’s hospital admission is authorized within minutes, allowing treatment to proceed promptly. The provider and patient are informed immediately. No human had to manually review, but the decision was based on solid data and policy – and is fully documented. The system moves to monitor if a **length-of-stay extension** request comes next.

### Case 2: Hospital Stay Extension Request (Approved)

**Scenario:** The patient has been in the hospital for 3 days. Near the end of the authorized period, the hospital submits a request to **extend the stay by 2 more days**, because the patient is still on IV antibiotics and not stable enough to discharge.

**Workflow Steps:**  
1. **Intake:** A new prior auth request (linked to the initial case) is received, asking for an extension from day 4 to day 5 of hospitalization. The request includes an updated physician note: patient continues to have fever, needs O2 support, so additional inpatient days are needed. IPAS ingests this update.  
2. **Data Update:** The agent pulls fresh data from the EHR – lab results and notes from day 3. Suppose it finds the patient had a spike in fever on day 2, and while improving, still requires oxygen at night. The policy for extensions says an extension can be approved if there is documented continued medical necessity (e.g., persistent symptoms or new complications).  
3. **Analysis:** The IPAS agent evaluates the criteria for extension. It sees that the patient’s condition, while slightly improved, still meets criteria for acute care (e.g., ongoing oxygen need). The initial approval was for 3 days; guidelines might allow up to 5 days for severe pneumonia if complications arise. In this case, the agent determines the extension of 2 days is **medically justified**. However, since this crosses the 3-day typical threshold, the agent’s confidence might be a bit lower than initial case (it’s a borderline scenario, but still within acceptable range).  
4. **Tentative Decision:** The agent recommends **Approval of the 2-day extension**. It prepares a rationale: “Patient still meets criteria for inpatient care on day 3 (e.g., still febrile and hypoxic), thus an additional 2 days (days 4-5) are warranted.”  
5. **Human Oversight (Optional in this case):** Depending on configuration, the system might flag this extension as a **medium complexity** case. Let’s assume IPAS’s confidence is high enough to approve without mandatory human review (perhaps this insurer trusts the AI for extensions within certain limits). The case appears on the dashboard, but marked as “Recommended: Approve (Auto)”. A utilization nurse could quickly glance at it in the table (see next section’s table) but doesn’t need to intervene because the rationale is sound. They might still use the chat to ask the agent a quick question like “Is the patient in ICU or regular floor?” and the agent would answer from EHR data. Satisfied, the nurse lets the auto-decision proceed.  
6. **Communication:** IPAS sends out an approval notification for the extension. It references the initial auth and now approves hospital days 4 and 5. An email goes out to the hospital’s case manager with this update immediately.  
7. **Logging:** The extension case is logged as approved. The memory notes that “extension of 2 days for pneumonia was granted; reason: continued hypoxia”. This could be useful if another extension is requested or for overall analytics (e.g., how often pneumonia stays get extended).

**Outcome:** The patient is allowed to remain hospitalized for the additional days needed. The process was slightly more involved than the initial case, but still largely automated. The human staff may have monitored it but did not have to manually gather data or justify the decision – the agent did that heavy lifting.

### Case 3: Hospital Stay Extension Request (Denied)

**Scenario:** After 5 days in hospital (initial 3 + 2 extended), the hospital submits **another request to extend the stay by a further 5 days** (days 6-10). In this hypothetical, the patient is clinically stable by day 5, but the hospital wants to keep the patient longer perhaps for observation or due to non-medical factors (e.g., waiting for a rehab bed). This is likely outside of guidelines.

**Workflow Steps:**  
1. **Intake:** The third request arrives, asking for days 6-10. The doctor’s note says the patient is much improved, vitals stable, oxygen discontinued, but “requesting extended stay for further physical therapy and monitoring.” IPAS ingests this.  
2. **Data Gathering:** The agent pulls the latest EHR info. It finds the patient is now stable: e.g., no fever, breathing ambient air, walking around. These findings indicate the patient **could be safely discharged to a lower level of care** (such as home with home health or to a rehab facility). The policy/guideline for pneumonia likely states that inpatient care beyond, say, 5-7 days requires a serious complication or failure of improvement – which is not the case here.  
3. **Analysis:** The agent immediately flags that criteria for further acute inpatient days are **not met**. It notes: the patient is stable, and the justification given (physical therapy) is not a valid reason for acute hospitalization (could be done outpatient or in rehab). The agent also considers if any other factors (social issues, etc.) might justify, but typically insurers focus on medical necessity. Here, it concludes this request **should be denied** as not medically necessary.  
4. **Confidence and Human Review:** Because a denial has higher implications (risk of patient harm if wrong, and likely to be contested), IPAS automatically sets this case to **require human review**. The agent is confident in a denial but, by design, any case that is borderline or a denial gets routed to an internal physician or senior reviewer for sign-off[[16]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Learn%20More). The system adds this case to the **priority review queue** (see next section) with a status “Action Required: AI suggests Denial”.  
5. **Internal Reviewer Chat:** A medical director or utilization review physician opens this case in the IPAS interface. They see the agent’s summary: *“Recommendation: Deny extension days 6-10. Patient is medically stable and does not meet criteria for continued acute hospitalization beyond day 5.”* The reviewer engages via the chat: “What criteria are not met? And what level of care is appropriate instead?” The agent responds, citing the policy (e.g., “Per MCG Pneumonia guidelines, inpatient discharge criteria met on day 5: patient afebrile 24h, O2 sat >95% on room air, ambulatory. No complications present to justify continued acute stay. Appropriate next level care: outpatient physical therapy or transfer to rehab facility.”). It may also highlight any data point: “No evidence of new infection or other issues in labs.” This thorough explanation, with references to guidelines, helps the human reviewer be confident the denial is justified.  
6. **Simulation (if needed):** The reviewer could use the /analyze\_and\_visualize command to double-check the agent’s decision process. The agent would show, for example, a decision flow where at the “medical necessity check” node the result was negative, hence leading to denial recommendation, and that a human review was triggered because the confidence threshold for auto-denial was not exceeded[[14]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=5,a%20detailed%20justification%20for%20the). This confirms that the proper checks were done and it wasn’t an arbitrary decision.  
7. **Final Decision:** The human reviewer agrees with the AI and clicks “Deny” to finalize. (If the human disagreed, they could override, but in this scenario they concur.) IPAS then prepares a **denial communication**. This includes a detailed denial letter/email to the provider, citing reasons and possibly suggesting alternatives or next steps (e.g., coverage for transfer to a rehab center might be mentioned, or the appeals process information is included). The justification uses the same points the agent explained: referencing the guidelines and patient data that support denial.  
8. **Communication:** The denial notice is sent out via email/fax to the hospital’s utilization team and is made available in the provider portal. It clearly states the request for days 6-10 is denied, and provides the clinical rationale and any instructions for appeal or further action.  
9. **Logging:** This case is logged as “Denied” with human review. All the conversation and rationale are stored. If the hospital appeals this denial, those records will be available for the appeals team (and indeed IPAS could also handle appeals by reviewing any new info, but that is beyond our current scope).

**Outcome:** The extension beyond medically necessary days is denied, avoiding unnecessary cost and inpatient days that don’t benefit the patient. The decision was made with a combination of AI analysis and human oversight, providing a clear audit trail and clinical justification. This protects against the scenario of blind AI denials – here the AI’s suggestion was vetted by a physician, aligning with good medical judgment.

These three cases demonstrate how IPAS handles different situations: a **clear approval**, a **moderate extension**, and a **complex denial**. In each scenario, the agent’s role and the human’s role adjust appropriately: - In Case 1, the AI handled everything end-to-end (autonomous). - In Case 2, the AI handled it but with a possibility of a light human check. - In Case 3, the AI did the analysis, but a human was in the loop to make the final call due to complexity.

All cases resulted in timely decisions (far faster than traditional manual review processes) and the communication to providers was near-instant and well-documented.

## Human-in-the-Loop and Case Management

To ensure safety and compliance, IPAS incorporates a **Human-in-the-Loop (HITL)** mechanism. Rather than leaving all decisions to automation (which, as surveys warn, can lead to overzealous denials[[3]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Health%20insurers%E2%80%99%20use%20of%20AI,judgment%20and%20exacerbate%20patient%20harm)), the system smartly triages cases and engages human reviewers where appropriate. The workflow includes a **case management dashboard** that prioritizes which cases need human attention.

In the IPAS dashboard, each incoming PA request is listed with key info and an AI recommendation status. For example:

| Case ID | Request Details | AI Recommendation | Human Review? | Priority |
| --- | --- | --- | --- | --- |
| 1001 | Initial Hospitalization (3 days) for pneumonia – Patient A | Approve (Criteria met)[[13]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=4,These%20criteria%20include) | **No** (Auto-approved) | Low |
| 1002 | Extension 2 days (days 4-5) – Patient A | Approve (Justified) | Optional (Auto unless flagged) | Medium |
| 1003 | Extension 5 days (days 6-10) – Patient A | **Deny** (Not medically necessary) | **Yes** (Review required) | High |
| 1004 | MRI scan request – Patient B | Partial Approve (MRI without contrast) | Yes (edge case) | High |
| 1005 | Outpatient PT sessions – Patient C | Approve (within limits) | No | Low |

*(Table: Example case queue with AI recommendations and human review status)*

In this table, we see how IPAS flags cases: - **No Human Review (Auto)**: Many routine cases (like 1001, 1005) are auto-approved/denied by AI. The “Human Review?” column is No. These might not even appear in a work queue for manual staff, except in an audit log. This dramatically cuts down the volume of work. For instance, if IPAS can auto-resolve even 50% of requests, that’s a huge efficiency gain. - **Optional or Deferred Review**: Some cases (like 1002) are auto-decided but kept visible for a quick post-check. Perhaps a nurse can scan them once a day just to ensure AI decisions look reasonable. These might be medium priority – the system is confident, but still logs them for oversight. - **Required Human Review**: Certain cases (1003, 1004) are flagged “Yes” for human review, meaning the AI will *not* finalize without a person’s input. Triggers for this could include: the AI’s confidence score is below a threshold, the outcome is a denial or partial approval, or the case involves high-risk/high-cost scenarios. These appear as High priority in the queue. The system ensures these get attention quickly (perhaps via notifications to the on-call reviewer).

This human-in-loop design ensures **no patient is denied care by AI alone without human eyes on it**, for those borderline or adverse decisions[[16]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Learn%20More). It also provides a safety net for any AI errors: the human can catch and correct them. Over time, as trust in the AI grows and it demonstrates accuracy, the threshold for auto-approvals might expand (like a “learning health system” approach).

**Reviewer Interaction:** For cases requiring review, the human uses the chat interface as described. They have all the AI’s compiled information at their fingertips (which is a big improvement over manually hunting through records). The agent effectively serves as a second pair of eyes, triaging information and even making a suggestion, but the human can agree or not. This setup addresses physicians’ concerns by keeping them in control for the tough calls, thereby fostering **trust in the AI recommendations rather than resentment or fear**.

**Documentation and Compliance:** Every decision, whether auto or human-reviewed, is documented with a **rationale**. IPAS automatically generates the clinical justification text to include in approval or denial letters, which the human can edit if needed. This standardized documentation reduces the time humans spend drafting decision letters by an estimated 70% (as seen with similar PA AI tools) and also reduces appeals since decisions are clearer[[17]](https://www.banjohealth.com/#:~:text=50)[[18]](https://www.banjohealth.com/#:~:text=less%20time%20drafting%20and%20updating,decision%20trees). In fact, by improving decision quality and explanation, IPAS aims to lower the appeal rate (currently about 20% of PAs end up in appeals[[19]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=delivery,survey.pdf)) – a metric we will track.

The system supports compliance with regulations and auditors by maintaining an **audit log** of how each decision was reached. If a regulator or an appeal committee asks “why was this denied?”, the insurer can produce the chat transcript and the criteria that were considered, showing a transparent process. This level of explainability is crucial for upcoming regulations and for clinician acceptance.

**Prioritization:** The queue is typically sorted by urgency (e.g., urgent inpatient requests vs routine ones) and by when a response is due (some regulations require decisions within 72 hours for urgent cases, etc.). IPAS can tag cases as “Urgent” vs “Standard” and ensure the turnaround times meet mandated deadlines. The AI helps expedite urgent cases especially – possibly even interrupting a reviewer only when necessary.

In summary, the **case management module** of IPAS allows the organization to handle high volumes of requests efficiently: the AI handles the bulk automatically, and the team of human reviewers can focus their expertise on the subset of cases that truly need their attention. This synergy leads to faster decisions (improving patient care by cutting delays) and less burnout on staff who previously spent endless hours on paperwork. The **balance of automation and human oversight** addresses both productivity and the ethical imperative of patient-centric care.

## Outcome Communication & Follow-Up

For each prior authorization determination, IPAS handles the **communication and next steps** seamlessly:

* **Automated Notifications:** Once a decision is finalized (approved, denied, or even partially approved), the system automatically notifies all relevant parties. Providers receive an email (or fax if needed) with the decision letter attached. Patients can also be notified (often patients get letters for denials). The use of Azure Communication Service (email/SMS) can automate sending these in a secure manner. The content includes patient identifiers, request details, decision, effective dates, and detailed reason for the decision. Automating this eliminates the current lag where staff might take a day or two to draft and send letters. It ensures near-real-time updates.
* **Partial Approvals:** In cases where IPAS provides a partial approval (e.g., approves a shorter length of stay than requested, or an alternate treatment), the communication clearly states that. For instance, “We have approved 2 days instead of the 5 days requested. This decision is based on XYZ criteria.” It may invite the provider to submit additional info if they believe more is needed.
* **Requests for Information:** If the AI cannot reach a confident decision due to missing info, it can generate an **RFI (Request for Information)** response. Rather than denying outright, IPAS might determine that key data is lacking (e.g., no lab results were attached). In such cases, it can compose a message: “Pending – need additional information: please provide recent lab results or justification for X.” This message can be sent to the provider, allowing them to respond and thus avoid a needless denial. This aligns with best practice to prevent repetitive resubmissions[[20]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=has%20mandated%20the%20Interoperability%20and,of%20documents%20is%20performed%20manually) – the agent can catch all missing pieces at once and ask for them proactively, instead of failing one check at a time. The system can pause the clock on the case until data comes, then resume the AI analysis once the provider responds (the chat interface could even allow providers a limited interaction to upload or explain additional info).
* **Integration with Provider Workflow:** The outcome can be integrated back into the provider’s workflow. For example, if the hospital uses an EHR that supports FHIR PA responses (as per the new CMS rules), IPAS can automatically send a FHIR PA Response message back through the API, updating the status in the provider’s system. This way, the doctor can see within their EHR that “auth is approved” without waiting for an email. This integration further speeds up the loop and fits into real-time point-of-care decisions[[21]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L239%20Until%20now%2C,of%20conversation%2C%20the%20organizations%20said)[[22]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=Until%20now%2C%20most%20attempts%20at,of%20conversation%2C%20the%20organizations%20said) (e.g., a physician can get an instant auth decision during a patient visit for something like an MRI – though our focus is hospitalization scenario, the principle applies).
* **Simulation for Patients:** In the future, patients might use a portal to simulate coverage as well (though not in current scope for IPAS). But the simulation engine could potentially be exposed in a limited way to providers: e.g., a doctor planning a treatment could use a UI to check “if I request X, is it likely to be approved?” by running through criteria. Highmark’s experimental real-time prior auth does something similar at point-of-care with checklists[[23]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=Abridge%27s%20prior%20authorization%20solution%20)[[24]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L196%20prior%20authorization,recommendations%20before%20anything%20is%20submitted), and our system design could accommodate that extension.
* **Final Documentation:** All communications are stored. Approved cases generate an authorization number and possibly update downstream systems (claims system is notified that this service is authorized, so when the claim comes it won’t be denied for lack of auth). Denials log an administrative code and can be used to auto-populate appeal forms if an appeal is opened later. By integrating IPAS with the claims and care management systems, it closes the loop in the insurance process.
* **Monitoring Outcomes:** Post-authorization, IPAS could optionally monitor outcomes. For instance, if a patient was approved for 5 days but got discharged in 3, that data could feed back (perhaps to refine future length predictions). If a denial led to an appeal, that triggers the feedback loop as described. These help measure the impact: e.g., faster decisions (target: decisions in <24 hours for 90% cases), reduced **appeals (target: e.g., cut appeals by 30% as Banjo Health saw)**[**[17]**](https://www.banjohealth.com/#:~:text=50)[**[18]**](https://www.banjohealth.com/#:~:text=less%20time%20drafting%20and%20updating,decision%20trees), and provider satisfaction.

By automating communication and follow-ups, IPAS ensures that once a decision is made, everything downstream happens quickly and correctly. This reduces the administrative burden on staff who would otherwise draft letters or manually update systems. It also improves transparency – providers are kept in the loop with clear reasons, which helps them accept the decisions or know how to respond (provide more info or plan discharge, etc.).

## Conclusion

IPAS (Intelligent Prior Authorization System) represents a **comprehensive, modern solution** to the prior auth challenge. By combining Azure’s powerful AI (GPT-4.1) with an agentic framework (Neurostack/LangGraph) and integrating deeply with healthcare data systems, IPAS can perform the heavy lifting that used to consume physicians and insurers with paperwork. It leverages **multi-source data** – from EHR clinical details to insurance rules – to make informed decisions, and uses memory and continuous learning to get smarter with each case. Crucially, IPAS is built with a **human-centric approach**: it provides explanations, simulation of its decision process, and involves human reviewers for the tough calls, thereby ensuring AI assistance **augments rather than overrides** clinical judgment[[3]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Health%20insurers%E2%80%99%20use%20of%20AI,judgment%20and%20exacerbate%20patient%20harm).

In our hospitalization prior auth use case, IPAS was able to approve needed care quickly (preventing treatment delays) and deny or modify requests when appropriate, all while documenting the reasons. The result is a win-win: patients get timely care, providers spend less time on admin and more on patients, and insurers operate more efficiently with consistent, evidence-based decisions. By reducing turnaround from days to minutes[[2]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L204%20The%20goal,into%20mere%20minutes%2C%20Rao%20noted) and cutting down repetitive manual checks, IPAS addresses the pain points of traditional PA processes. Furthermore, its compliance with standards (FHIR, CMS rules) and thorough logging means it’s ready for the future of healthcare interoperability and audits.

In summary, IPAS is an **agentic AI system** that embodies “intelligent prior authorization.” It demonstrates how AI can transform a once tedious process into a streamlined, transparent workflow – improving outcomes for all stakeholders while maintaining the necessary safeguards in such a critical area of healthcare. With IPAS, the prior authorization process becomes not a roadblock, but a smart guardrail that ensures appropriate care with minimal friction.

[[1]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=in%20late%202024%20found%20that%2C,completing%20those%20requests%20each%20week) [[2]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L204%20The%20goal,into%20mere%20minutes%2C%20Rao%20noted) [[12]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L222%20that%20provides,based%20clinical%20guidelines) [[21]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L239%20Until%20now%2C,of%20conversation%2C%20the%20organizations%20said) [[22]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=Until%20now%2C%20most%20attempts%20at,of%20conversation%2C%20the%20organizations%20said) [[23]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=Abridge%27s%20prior%20authorization%20solution%20) [[24]](https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai#:~:text=match%20at%20L196%20prior%20authorization,recommendations%20before%20anything%20is%20submitted) Highmark, Abridge team up to design AI-powered prior auth tech

<https://www.fiercehealthcare.com/health-tech/highmark-health-taps-abridge-build-real-time-prior-authorization-using-ai>

[[3]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Health%20insurers%E2%80%99%20use%20of%20AI,judgment%20and%20exacerbate%20patient%20harm) [[16]](https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials#:~:text=Learn%20More) How AI is leading to more prior authorization denials | American Medical Association

<https://www.ama-assn.org/practice-management/prior-authorization/how-ai-leading-more-prior-authorization-denials>

[[4]](https://www.banjohealth.com/#:~:text=Effortless%20Integration) [[8]](https://www.banjohealth.com/#:~:text=Purpose) [[17]](https://www.banjohealth.com/#:~:text=50) [[18]](https://www.banjohealth.com/#:~:text=less%20time%20drafting%20and%20updating,decision%20trees) Banjo Health | AI-Powered Prior Authorization Software

<https://www.banjohealth.com/>

[[5]](https://www.nexastack.ai/blog/agentic-ai-agent-framework#:~:text=Enterprise%20AI%20agents%20are%20autonomous,improvement%20over%20time) AI Agent Framework: Strategic Implementation

<https://www.nexastack.ai/blog/agentic-ai-agent-framework>

[[6]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=2,agents) [[7]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=11,agent%20performs) [[9]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=1,survey.pdf) [[13]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=4,These%20criteria%20include) [[14]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=5,a%20detailed%20justification%20for%20the) [[15]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=To) [[19]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=delivery,survey.pdf) [[20]](https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai#:~:text=has%20mandated%20the%20Interoperability%20and,of%20documents%20is%20performed%20manually) Transforming Prior Authorization for Insurers via Agentic AI | HCLTech

<https://www.hcltech.com/blogs/revolutionizing-prior-authorization-insurers-agentic-ai>

[[10]](https://www.precisionaq.com/hubfs/5014803/Modernizing%20PA%20Process.html#:~:text=The%20Catalyst%3A%20CMS) [[11]](https://www.precisionaq.com/hubfs/5014803/Modernizing%20PA%20Process.html#:~:text=The%20Great%20Migration%3A%20From%20X12,to%20FHIR) Architecting the Future of Prior Authorization | Interactive Report

<https://www.precisionaq.com/hubfs/5014803/Modernizing%20PA%20Process.html>