ChatGPT

LLM-Based Automation in Air Traffic Control: Feasibility, Architecture, and Implementation

Introduction

The U.S. air traffic control (ATC) system faces a significant workforce shortfall, with around 10,600 certified controllers as of late 2023 – roughly 2,000 below the FAA's own staffing targets and over 4,000 below joint FAA/NATCA optimal levels [1]. This **3,000+ controller deficit** strains the National Airspace System (NAS) and has prompted exploration of automation or AI augmentation to maintain safety and efficiency. Recent advances in *agentic* large language models (LLMs) – AI systems that can reason, plan, and use tools autonomously – suggest it may be feasible to offload certain controller tasks to AI, thereby easing workload or even fully automating some control functions. However, **air traffic control is a safety-critical domain**, and any AI-based solution must meet stringent real-time performance, reliability, and regulatory requirements.

This report investigates the feasibility, proposed architecture, and implementation path for an LLM-based "digital controller" agent to automate or assist ATC operations in U.S. domestic airspace. Key considerations include: the scope of tasks that could be fully automated versus those requiring human oversight, integration of the AI with radar/surveillance, weather, and communication systems in real time, technical architecture (LLM with retrieval-augmented memory, planning modules, and external tool use), regulatory and certification challenges under FAA rules, the current state of AI in air traffic management (ATM) research and trials, and human factors and safety constraints. We provide a comprehensive analysis with technical feasibility assessments, regulatory analysis, risk summaries, and comparative tables of possible architectures and deployment models. The goal is to outline how an **"AI air traffic controller"** could be developed and deployed to help close the controller staffing gap while upholding or enhancing the safety and throughput of U.S. airspace.

Feasibility of Automating Air Traffic Control Tasks

Automating the full breadth of air traffic controller tasks is an **extremely ambitious goal**. Controllers perform diverse functions: monitoring aircraft positions, ensuring separation minima, issuing clearances and instructions via radio, sequencing arrivals/departures, coordinating with adjacent sectors, managing emergencies, and more. Many of these tasks require complex real-time decision-making, situational awareness, and communication skills that have traditionally been **beyond the reach of AI** [2]. The advent of modern AI offers both great promise and serious risks: as one expert put it, introducing AI into the control tower "could mark a step-change improvement in aviation safety, or else could usher in a flush of 'AI-induced' accidents" if not done carefully [2]. Contemporary AI systems (including LLMs) have known weaknesses – from data bias to unpredictability on edge cases – that must be overcome before they can safely shoulder frontline ATC responsibilities.

Augmentation vs. Full Automation: In the near to medium term, a more feasible approach is deploying AI as a **decision-support "wingman"** to human controllers, rather than a full replacement. AI can

continuously analyze data and traffic patterns in the background, offering *instant insights* and suggestions to assist overloaded controllers [3]. For example, an AI system with a wide view of all traffic could *furnish early conflict warnings or optimal sequencing recommendations* for flights approaching a busy airport [3]. Human controllers would remain the final decision-makers, but the AI would act as a real-time assistant – reducing cognitive load and helping each controller safely manage more aircraft. Michael McCormick, a former FAA air traffic chief, predicts that "controllers would still make the decisions, but AI would provide alerts and recommendations" to them as the "next leapfrog in technology" for ATC [4]. This teaming paradigm aims to **offset the labor shortfall by boosting each controller's capacity**, rather than outright replacing humans. It is also more palatable from a safety and regulatory standpoint in the short term, since the AI's suggestions can be vetted by a human before execution.

Looking further ahead, **full automation** of certain ATC roles may become achievable as AI reliability improves. This might first occur in constrained environments – e.g. managing en-route cruise traffic under instrument flight rules (IFR) in high-altitude sectors, or controlling a quiet regional airport tower during off-peak hours with an AI "digital tower" system. Such scenarios have more predictable patterns and fewer simultaneous conflicts, making them potential early candidates for autonomy. Indeed, research prototypes have shown that an LLM-based agent can *resolve simple air traffic conflicts without human intervention* in simulation [5] [6]. However, **fully autonomous ATC in busy, complex airspace remains a daunting challenge**. The system would need to handle dense traffic, unpredictable pilot behavior, mixed VFR/IFR operations, weather disruptions, and emergency situations – all with **zero tolerance for serious errors**. At present, **no AI meets the level of judgment, reliability, and real-time adaptability** that certified human controllers provide [7] [8]. Regulators in Europe have even stipulated that generative AI (large LLMs) are *"not allowed… at least not now in the current legal framework"* for live ATC operations [8] due to safety concerns. Thus, full automation is likely a long-term prospect that will require incremental progress and extensive validation.

To weigh these approaches, **Table 1** compares an augmented human-AI system versus a hypothetical fully autonomous ATC system across key criteria:

Criteria	AI-Augmented ATC (Human-in-the-Loop)	Fully Autonomous ATC (Human-on-the-Loop)
Human Role	Controllers make final decisions; AI provides suggestions, alerts, and data insights [4]. Human approves AI recommendations before action.	AI agent makes control decisions and issues clearances directly; humans supervise multiple sectors or intervene only if needed (monitoring role).
Tasks Automated	Routine or time-consuming tasks (e.g. scanning for conflicts, reading back pilot requests, data entry, updating flight strips). AI assists with planning optimal sequences, detecting deviations, etc. [3].	Almost all controller tasks automated: monitoring radar, conflict detection/ resolution, communications with aircraft, coordination with other facilities. Humans handle exceptions or override as needed.

Criteria	AI-Augmented ATC (Human-in-the-Loop)	Fully Autonomous ATC (Human-on-the-Loop)
Technical Feasibility	High feasibility in near term. Narrow AI/ML tools already excel at specific subtasks (e.g. speech recognition, conflict prediction). LLMs can emulate standard phraseology and suggest resolutions [9]. Human oversight mitigates AI errors.	Low feasibility currently. Requires human-level reasoning, complex scenario handling, and extreme reliability. Current AI lacks full situational awareness and judgement for high-density operations [2] [10]. Would need breakthroughs in AI robustness and extensive training on real ATC scenarios.
Safety and Certification	Easier to certify as a *Decision Support Tool*. AI errors are caught by humans, so system can tolerate occasional mistakes. Regulatory approval still needed for specific functions (e.g. alerting logic), but risk is lower with human in loop.	Very stringent safety assurance required. Would likely need certification akin to avionics at Design Assurance Level A (catastrophic risk). Every possible failure mode must be mitigated. Achieving regulator trust in an unmanned ATC system would require exhaustive testing, simulation, and gradual trials in low-risk environments.
Impact on Workforce	Increases productivity of each controller – potentially allowing staffing of more sectors or extending operating hours with the same workforce. Reduces stress and fatigue, improving retention [11]. Does not eliminate need for human controllers but narrows the shortfall.	Could directly fill positions that are currently vacant, theoretically alleviating the 3,000-controller gap if fully realized. However, in practice a human backup might still be required for each AI-controlled sector, at least initially, limiting net workforce reduction.
Timeline to Deploy	**Short-term (1–5 years)** for initial capabilities. Prototypes already in testing (e.g. voice transcription assistants, AI conflict advisors). Gradual deployment possible after safety certification of limited-scope tools.	**Long-term (10+ years)** for high-density airspace. Some autonomous tower or low-traffic en-route operations might be achievable sooner (5–10 year range) if technology progresses and regulatory changes allow. Full nationwide autonomous ATC would likely span decades.

Table 1: Comparison of AI-Augmented vs. Fully Autonomous ATC Approaches.

In summary, **augmenting human controllers with AI assistance appears to be the most practical path in the near term**. This can address the labor shortfall by enabling each controller to handle more traffic with less stress, effectively multiplying workforce capacity. Full automation is an aspirational goal for specific use cases, but it will require major advances in AI capability and extensive safety proof. The remainder of this report focuses on an implementation approach that begins with human-AI teaming and incrementally moves toward higher autonomy as technology and trust mature.

Real-Time Performance and System Integration

Any AI-based ATC solution must function within the **real-time constraints** of the NAS. Controllers operate on a second-by-second basis – for example, noticing an emerging conflict and issuing a course correction within moments to maintain separation. An AI "controller" would need to ingest high-volume, streaming data from radar and surveillance systems, analyze complex air traffic situations continuously, and produce correct instructions or alerts *without undue delay*. The FAA's recent outage at Newark, where a radar blackout for just 90 seconds caused a major disruption, underscores that *even a minute-long lapse is unacceptable* in ATC operations [12]. Thus, system architecture and performance engineering are critical to meet strict latency and uptime requirements.

Integration with National Surveillance and Data Systems: The U.S. NAS already provides rich machine-readable data streams that an AI agent can leverage. Through the FAA's System Wide Information Management (SWIM) network, a controller automation system could subscribe to real-time feeds of aircraft track data (radar and ADS-B positions), flight plans, weather information (e.g. NEXRAD radar, Terminal Area Forecasts), and aeronautical data [13] [14]. Modern ATM platforms like *Flyways AI* demonstrate that integration of over 100 data feeds (including SWIM and weather services like CoSPA) is feasible, creating a live 4D digital twin of the airspace [15] [16]. In fact, Flyways AI's architecture shows how an open data integration layer can fuse geospatial data (traffic, weather, etc.) and feed it into predictive models in real time [17] [18]. An agentic LLM-based controller would require a similar **data fusion module** that continuously updates the AI with the current state of the world. This could be implemented via middleware that pulls SWIM messages (track updates, metering info, NOTAMs, etc.) and translates them into a format the AI can process (for instance, a structured summary or a prompt template). Ensuring **low latency** in this pipeline is vital – updates might need to flow at 1–2 second intervals for fast-moving situations.

Communication System Interface: Air traffic control is as much about communication as surveillance. Controllers talk to pilots via radio (voice) and increasingly via digital link (e.g. CPDLC for en-route clearances). An AI system must be able to both *understand incoming pilot requests and transmit outgoing instructions* in standard phraseology. Recent projects have achieved near-real-time transcription of pilot-controller voice communications using AI speech recognition, feeding the text into an LLM for analysis [19]. For example, **MosaicATM's Sky+AI prototype** uses a two-step pipeline: first, speech-to-text converts radio calls to text; second, a domain-tuned LLM analyzes the transcript to detect anomalies or compliance issues and can even suggest corrective actions [19]. This capability would be foundational for an AI controller: it must reliably **parse pilot read-backs, requests, and reports**. Conversely, the AI must generate clear and correct voice outputs. LLMs excel at natural language generation; with fine-tuning on ATC phraseology corpora, an LLM can learn to produce instructions in the precise format controllers use [9]. There are already demonstrations of LLMs that, given a command intent, will output a proper ATC clearance or advisory. For instance, a fine-tuned model can take an input like "Envoy four-one-nine-two, runway three six right, wake turbulence caution… line up and wait" (pilot call) and respond as a tower controller: "Envoy 4192, caution wake turbulence, runway 36 right, **cleared for takeoff**" – matching official phraseology and including required warnings [20]

Context utterance (input to the agent)	Response utterance
state line-wait - id envoy four one nine two - runway three six right - wake turbulence - crew okay line up and wait envoy four one nine two	controller envoy four one nine two caution wake turbulence runway three six right cleared for takeoff

Figure 2. Sample controller answer of model fine-tuned to answer as a controller or pilot in the local controller environment

. *Figure 1* illustrates an example of an LLM (trained on ATC dialogues) generating a controller's response given the context of a pilot's call. This shows that the **communication latency** can be low – essentially the time to run an LLM inference (a few hundred milliseconds on modern hardware) plus radio keying – and the style and clarity can meet operational standards with proper training.

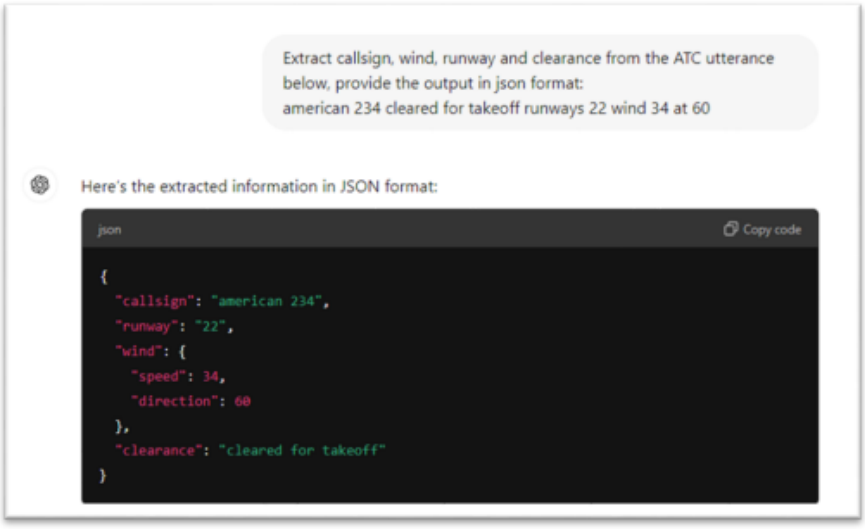

Figure 1: Sample JSON output using an LLM to parse and reformat

Figure 1: An example of an LLM parsing a spoken ATC clearance into structured data (JSON). This demonstrates how the AI can convert unstructured pilot/controller utterances into machine-readable form for downstream processing.

Beyond voice, integration also extends to **digital communication channels**. An AI controller could interface with systems like **CPDLC (Controller Pilot Data Link Communications)** to send text clearances to suitably equipped aircraft. This would actually simplify matters, as the AI can exchange structured messages directly without speech recognition in those cases. The system would tie into existing communication infrastructure via APIs, ensuring any instruction the AI "sends" is delivered to pilots either as a synthesized voice on the frequency or as a text message in the cockpit.

Real-Time Processing and Hardware: Achieving the necessary throughput will likely require **on-premises deployment** of the AI system at control facilities or major data centers with direct NAS data links. Given concerns about data security and the need for instantaneous responses, cloud-based solutions are less suitable for real-time control (though cloud can be used for offline training). MosaicATM's feasibility study emphasizes the value of on-premise AI for aviation data privacy and latency reasons [21]. The AI models

(especially if using large neural networks) will need acceleration via GPUs or specialized AI chips to handle the continuous stream of inputs and outputs. However, one advantage is that the **task can be highly parallelized**: monitoring dozens of aircraft and radio channels simultaneously is something AI can do without fatigue, as long as it has sufficient computing resources. A possible architecture might involve a cluster of inference servers subscribed to all relevant data feeds and radio channels for a given airspace sector, with each server running instances of the LLM agent that focus on different tasks (e.g. one focusing on conflict detection, one on communications, etc., or even one agent per active flight if needed). The system must also operate under **real-time operating constraints**, possibly with a priority scheduler to ensure critical tasks (like conflict alerts) are processed immediately, whereas less critical tasks (like routine traffic flow optimization) can use spare cycles.

Finally, **system reliability and redundancy** are paramount. Just as current ATC automation systems (like radar displays and flight data processors) have backups, the AI system would need failsafe mechanisms. This could include a hot-standby instance of the AI agent that takes over if the primary fails, or reversion to manual control (with human controllers) if any anomaly is detected in the AI's function. Downtime or lag in an AI-driven ATC system could be catastrophic, so integration testing must prove that the AI can meet the **99.999% availability** and strict response time requirements typical of ATC equipment. This is part of the broader safety assurance discussed later, but it is clear that *high-performance computing and robust engineering are as critical as the AI algorithms themselves* for this application.

Agentic LLM Architecture for ATC Automation

The proposed technical architecture leverages an **agentic LLM** at its core, augmented with specialized modules for memory, planning, and tool use. The concept is to imbue the LLM with the ability to *perceive the air traffic environment, reason about it, and take actions (or recommend actions) in a closed-loop fashion*. Several research prototypes in ATM have already explored such architectures, offering a template for how to design a safe and effective AI controller. **Figure 2** below outlines the high-level structure of an LLM-based ATC agent, as synthesized from recent literature [22] [23]:

- **Perception & State Integration:** This front-end ingests live data (radar tracks, weather, etc.) and relevant static context (airport configurations, airspace maps, procedures). Data may be converted into textual or structured summaries that the LLM can interpret. For example, a surveillance processor might continuously generate a list of all aircraft in the sector with their callsigns, positions, altitudes, and speeds – the LLM could take this as part of its context window or query it via a tool function. Similarly, transcribed radio messages and flight plan info feed into the agent's context. By giving the LLM an up-to-date "picture" of the airspace, we ensure it has the situational awareness needed for decision-making.

- **LLM Core (Reasoning Engine):** At the heart is a large language model (such as a GPT-style transformer, potentially fine-tuned on aviation data). This core is prompt-driven: we provide it with an initial system prompt that encodes general instructions (e.g. "You are an air traffic controller AI. Ensure safe separation (5 NM horizontal, 1000 ft vertical) at all times. Follow FAA phraseology. etc.") along with the dynamic state from perception. The LLM then *generates outputs in natural language* – either directives to aircraft ("Delta 123, turn right heading 250 to avoid traffic") or internal reasoning steps. Crucially, this LLM is equipped with **agentic capabilities**: it can plan multi-step solutions and invoke external tools. Rather than a fixed input-output mapping, the LLM runs iteratively, using a

chain-of-thought approach where it can contemplate the situation, consult additional information, and decide on actions.

- **Tool Use and External APIs:** A defining feature of an agentic LLM is the ability to call external functions to extend its capabilities beyond text generation [22] [24]. In the ATC context, we incorporate a suite of **"ATC tools"** the LLM can invoke. These are essentially API endpoints or function calls that perform specific queries or actions in the environment:

- *State Query Tools:* e.g. `GetAllAircraftInfo()` returns a structured list of all aircraft and their kinematic states [25]; `GetConflictInfo()` returns pairs of aircraft that are in conflict or predicted loss of separation, including metrics like time to closest approach [26]; `GetWeatherInfo()` might retrieve current storm cell locations or winds; etc. These allow the LLM to get precise, up-to-date data rather than relying solely on its textual memory.
- *Action Tools:* e.g. `SendCommand(Callsign, Instruction)` which interfaces with the radio/CPDLC system to issue a clearance to an aircraft [27]; `HandoffSector(Callsign, SectorID)` to transfer communication to another sector's AI/human; or perhaps `AdjustFlightPlan(Callsign, route change)` to file a new route in the system. When the LLM decides on an action, it uses such a tool, which then affects the simulated or live environment (and the results loop back via the state updates).

- *Knowledge Tools:* e.g. `SearchProcedures(query)` to fetch relevant FAA regulations, SOPs or approach plates from a database (this is a form of **retrieval-augmented generation**, ensuring the LLM can reference authoritative data on demand). Another key tool is the **experience memory** described next.

- **Experience Library (Retrieval Memory):** To improve long-term reasoning and avoid repeating mistakes, the architecture includes a **vector database** of past scenarios and solutions – essentially an LLM-accessible memory bank [28] [29]. After the agent handles a scenario (for example, resolving a conflict between two aircraft), it can store the key details and the outcome as an "experience document" in this repository [30]. The next time the agent faces a similar situation, it can query this library (via a tool like `SearchExperienceLibrary(conflict_pattern)`) to retrieve what it learned before [31]. This is akin to a junior controller learning from past cases. For instance, if it previously solved a converging traffic conflict by issuing speed adjustments, that experience can be surfaced to inform the resolution of a new, analogous conflict. The use of a vector database (with embeddings of the scenario description) allows semantic search for relevant experiences, even if the exact aircraft or altitudes differ [28]. This **retrieval-augmented approach (RAG)** can both improve performance (by reminding the LLM of successful strategies) and provide a form of **explainability** (since the AI can cite a similar past case as rationale for its suggestion).

- **Multi-Agent Structure (Planner/Verifier):** To enhance safety, the architecture can be expanded into multiple LLM agents with different roles. Recent research introduced a *planner-verifier-executor* model [32]. In this setup, one LLM agent (Planner) looks at the situation and proposes a set of actions (a plan) to resolve conflicts or manage traffic. A second LLM agent (Verifier) then reviews this plan against safety criteria – essentially playing the role of a watchdog or junior supervisor. The verifier might internally simulate the outcome (using the same tools to project conflicts) and ensure no new conflicts or rule violations will occur. Only if the plan passes this check does the executor agent actually carry out the commands in the live system [33] [27]. This layered approach adds redundancy

and helps catch errors: the verifier provides an independent analysis with the ability to veto or adjust the Planner's solution if it foresees a problem. In testing, such multi-agent configurations showed higher success rates in conflict resolution than a single agent alone [34], precisely because the second agent caught corner cases the first might miss. This concept mirrors human ATC practices like requiring two controllers to sign off on certain critical clearances or having a supervisor monitor operations.

- **Human Interface:** Even in a fully autonomous mode, a human-over-the-loop interface is essential. The AI agent should present its decisions and the rationale in a form that human overseers (or pilots, when applicable) can understand. One of the advantages of using LLMs is their ability to produce **natural language explanations** for their actions [34]. Unlike a black-box algorithm, an LLM agent can output a rationale like: *"I am vectoring Flight X off its path to avoid conflict with Flight Y, based on their converging trajectories. The turn is 30° right which will establish the required 5 NM separation in 2 minutes."* These explanations, logged and shown on a dashboard, greatly aid transparency and trust. Controllers or traffic managers monitoring the system can see *why* the AI is doing something and intervene if that reasoning seems flawed. This design principle aligns with the SESAR program's emphasis on eXplainable AI in ATC tools (e.g. the TAPAS project sought to make AI decisions accessible to controllers via visual analytics [35]).

In practice, implementing this architecture would involve using an **open-source LLM** (for data controllability and on-prem deployment) fine-tuned on aviation-specific data, plus a lot of surrounding code for the tools and interfaces. The Delft University team's recent agent, for example, combined a pretrained LLM with Python-based tools connected to the BlueSky ATC simulator [22]. The LLM's prompts were carefully structured to integrate tool outputs and maintain dialogue state (chat history and scratchpad memory) [36] [24]. Their results were promising: the best configuration *solved 119 out of 120* mid-air conflict scenarios (with up to 4 aircraft) autonomously and provided human-level textual explanations of the resolution strategies [6]. This was achieved by the LLM agent dynamically querying the simulator for aircraft data, identifying conflicts, and issuing resolution commands via the defined tools, all while learning from each interaction by updating its experience library [25] [30]. Such findings suggest that the core reasoning and closed-loop control is technologically within reach, *at least for simplified scenarios*. The key challenges are scaling this up to the complexity of real airspace and integrating the rigorous safety guardrails required.

Ensuring Safe Behavior: As part of the architecture, it's critical to embed *hard constraints and validation checks*. An AI agent must never violate fundamental safety rules (e.g., minimum separation standards or not clearing an aircraft to an altitude if another is there). Some of these can be enforced by the tools themselves – for example, an action tool could refuse to issue a clearance that would immediately breach separation, or an external conflict probe could run after every AI command to double-check that no conflicts result. The architecture could include a rule-based engine running in parallel with the LLM: the LLM proposes actions, and a separate safety module (coded with deterministic rules/regulations) validates them before execution. This hybrid approach (AI plus rule-based safety net) is suggested by experts who note that an AI for ATC should *"have curated data and strict rules, similar to the software that controls autonomous vehicles, to eliminate mistakes"* [37]. In essence, the LLM provides flexibility and reasoning, while a rule-based overlay ensures compliance with inviolable constraints.

In summary, the agentic LLM architecture for ATC involves a **network of components orchestrated around an LLM's cognitive ability**: real-time data feeds to perceive the environment, tool APIs to act on and query the environment, a memory/knowledge base to learn from experience, multi-agent or

hierarchical decision layers for validation, and fail-safes to enforce safety. This design draws on state-of-the-art AI research while acknowledging the need for domain-specific safeguards. It aims to capture the *human-like situational reasoning* of an expert controller within a reproducible, testable software system.

Regulatory and Safety Requirements

Deploying an AI controller in U.S. airspace faces stringent **regulatory hurdles**. The Federal Aviation Administration (FAA) has ultimate authority over ATC operations and equipment, and their prime directive is safety. Any system that issues clearances to aircraft (even under human supervision) is effectively making decisions that could affect life and property; as such, the FAA will treat it as a safety-critical system requiring rigorous certification. Unlike some tech domains that adopt "move fast and break things," aviation demands *"move carefully and prove it works (then triple-check it)"*.

FAA Certification and Policy Context: The FAA has begun grappling with how to certify AI-driven systems. In 2023, the agency released its first *"Roadmap for AI Safety Assurance"*, outlining principles for introducing AI in aviation [38] [39]. The FAA emphasizes working within existing safety frameworks: **existing regulations (e.g. 14 CFR parts 23, 25 for aircraft systems) are largely performance-based, requiring that any system perform its intended function without introducing unacceptable hazards** [40]. This means an AI ATC system would need a thorough safety case demonstrating it meets or exceeds the safety level of current operations. Notably, the FAA advocates an **incremental approach** – integrating AI in limited, low-risk applications first, learning from those, and gradually expanding usage [41] [42]. For ATC, this could translate to initially using the AI as a non-controlling decision support tool (no direct authority, just recommendations), then maybe allowing it limited control in specific scenarios (e.g. a training simulator or shadow operations), before any live autonomous control. The FAA is also participating with NASA in developing certification methodologies for AI, acknowledging that traditional deterministic certification doesn't directly apply to learned behaviors [43] [44].

One key regulatory question is under what framework an AI controller would be certified. Existing ATC automation (like the Traffic Collision Avoidance System *TCAS* or airport Surface Movement Guidance systems) are certified via the FAA's Acquisition Management System for NAS equipment, often using standards like DO-278A (guidelines for ground/ATC software). An AI system might need a new or supplemental set of standards. The FAA's AI Roadmap suggests using industry consensus standards and focusing on *safety assurance of the AI's training data, algorithms, and performance* [39] [45]. Likely, metrics like **mean time between hazardous failures** will need to be defined for AI decisions. The FAA will also require that the AI's developer assume responsibility for the system's performance and be able to explain and justify it – a challenge since learned systems are less transparent by nature [46] [47]. Techniques like LLMs providing natural language rationales and having the aforementioned rule-based checks can help bridge that explainability gap, making it easier to assure regulators that the AI behaves in a predictable, bounded way.

Airspace Classes and Operational Approvals: U.S. domestic airspace ranges from busy Class B terminals to quiet Class G uncontrolled areas. The level of approval needed may vary by where the AI is deployed: - In **Class A (high-altitude en-route)** and **Class B/C (busy terminal)** airspace, ATC services are absolutely critical and tightly regulated. Any AI controlling traffic here would likely require a **certified system status** equivalent to current en-route automation systems. The FAA might start by approving AI assistance for en-route sectors with low traffic at night, for example, before anything in busy daytime skies. - **Class D (small towered airports)** or **remote tower operations** might be a testbed. The FAA could allow an "AI Controller"

at an airport that currently has limited hours or frequent personnel shortages, under strict monitoring. There is precedent for *remote digital towers* (using cameras and sometimes assistive automation) in places like Europe – the U.S. could similarly trial an AI-driven tower in low-risk conditions (good weather, low volume) as a pilot program. - **Uncontrolled airspace (Class G)**: While no ATC service is provided here normally, for research purposes AI could manage drone traffic (UTM systems) or act as an advisory service to see how it performs without risk to current operations. NASA's Unmanned Aircraft System Traffic Management experiments, for instance, allowed a high degree of automation in segregated airspace for drones.

A phased certification might involve issuing ** waivers or experimental certificates** for AI operations initially. Similar to how autonomous drone flights get waivers of certain FAA rules, an AI ATC agent might operate under an "Experimental ATC Facility" status, with tight constraints on what it can do (perhaps it can only control specially equipped test flights, or only issue advisory altitudes that pilots are not mandated to follow). This could evolve into a more formal certification if the trials prove safe.

Safety and Redundancy Requirements: No matter the airspace, any operational deployment will demand that the AI system *demonstrate reliability on par with or better than human controllers*. Currently, the ATC system's safety record is extraordinarily high – serious incidents are very rare given millions of operations. To not degrade this, the AI would be expected to have **error rates well below those of humans**. This includes not only making correct decisions but also maintaining function under duress (no crashes or unplanned downtime) and resisting cyber threats. Cybersecurity is a regulatory focal point; an AI controller connected to networks and potentially vulnerable to hacking or data poisoning must have robust protections (authentication, encryption, etc.) as it now becomes part of critical infrastructure.

From a **policy perspective**, the FAA and industry will also weigh the *human ramifications*. There are likely to be labor relations considerations (NATCA, the controller's union, will rightfully demand proof that any AI won't compromise safety or lead to unreasonable staffing practices). The FAA will have to engage stakeholders, including airlines and pilots, to ensure confidence in the system. Pilots, by regulation, must obey ATC instructions – but will they trust instructions that they know are coming from an AI? To ease this, initial deployments might be **transparent about AI assistance** (e.g. "Advisory AI System" labels) and gradually integrate such that to pilots, nothing changes in terms of procedure (they still talk to a "controller," even if it's synthetic). Over time, regulations may need updating – for instance, the FAA might one day revise CFR provisions that currently assume a human controller. But in the interim, a likely approach is to **keep a human in ultimate charge** (the human can always override the AI, and the AI might be officially considered a decision support tool rather than an independent controller in legal terms).

Global and Legal Considerations: Internationally, any change to ATC practices involves ICAO (International Civil Aviation Organization) standards and coordination with other countries for flights crossing boundaries. If the U.S. were to adopt AI in ATC, it would likely do so in concert with ICAO working groups or at least inform them, to ensure harmonized procedures. Europe's regulations, especially the pending EU AI Act, may classify autonomous ATC as "high-risk AI" requiring specific conformity assessments (the EU already hinted that GenAI is not permitted in ATC until safety can be proven [8]). These global regulatory currents will influence how the FAA proceeds. It's possible the U.S. could take a pioneering role if it can solve the safety case, or conversely, if Europe finds a safe methodology first, the FAA might adopt similar guidelines.

In short, **regulation is perhaps the tallest hurdle** for an AI ATC system. Meeting it will require meticulous engineering (to create a demonstrably safe system), extensive simulation and shadow-mode testing (to

gather evidence of safety), and a collaborative approach with regulators (to define new standards where needed). The good news is that FAA leadership has shown openness to AI as a support tool: their research plan explicitly includes using AI for controller decision support and other areas [48]. The key will be convincing them – and the flying public – that an AI controller can enhance safety rather than threaten it. The path to certification will likely involve gradually increasing the AI's level of autonomy, as described in the next section on implementation.

Current State of AI in Air Traffic Management

To gauge feasibility, it's instructive to examine what has already been achieved in academia and industry regarding AI for ATM. In recent years, there has been **rapid progress in applying AI/ML to various ATC-related tasks**, though most projects stop short of full automation. Below we summarize relevant efforts:

- **SESAR Projects (Europe):** Europe's SESAR Joint Undertaking (analogous to the FAA's NextGen) has funded multiple exploratory projects on AI in ATC. The *AISA project* (Artificial Intelligence Situational Awareness) developed knowledge-graph and ML techniques to give a virtual assistant a form of "situational awareness" of air traffic, aiming to support controllers with traffic predictions [49]. *TAPAS* (Transparent ATC Autonomous System) focused on **explainable AI** – it built AI agents whose decision processes were fed through visual analytics tools to make them understandable to controllers in simulations [50]. Another project, *ARTIMATION*, worked on visualizing AI reasoning to increase transparency [51]. These projects, tested in human-in-the-loop simulations, found that while AI can handle certain tasks, there was still a noticeable gap between AI and human controllers in true **situational awareness and nuanced decision-making** [52]. This motivated the push for more human-like reasoning AI (hence interest in LLMs post-2023). It's worth noting these were research trials – no SESAR AI system has been deployed operationally yet, but they paved the way in identifying challenges (like the need for explainability and trust).

- **LLM-Based ATC Research:** The most cutting-edge research specifically applying LLMs to ATC has emerged only in the last year or two. A notable study by Andriuškevičius & Sun (2024) introduced a *"large language model embodied agent"* for air traffic control [5]. They integrated GPT-style models with a simulator (BlueSky) via function-calling, as described earlier, and even introduced the concept of an experience replay memory [53] [28]. Their agent could autonomously resolve imminent conflicts among aircraft and explain its reasoning in plain language [34]. This is a breakthrough because it shows an AI handling a core controller task (separation assurance) in a realistic setting. However, it was still limited to scripted scenarios with a handful of aircraft. Another example is *MosaicATM's Sky for All* study (2025), which, while focusing on voice communication analysis, also explored fine-tuning LLMs on ATC dialogue so they can *"communicate as a pilot or controller"* in simulation [9]. In one use-case, they integrated a fine-tuned LLaMA model as a "pseudo-pilot" in a training simulator, which could respond to trainee controllers with realistic behavior and even generate off-nominal (emergency) events to test them [9]. This shows LLMs are capable of *learning the language and protocols of ATC to a degree that they can simulate the role of ATCO or pilot* – a good proxy for eventually performing the real role.

- **Voice Recognition and NLP for ATC:** Outside of decision-making, AI has made strides in automating communication and data analysis. **MITRE** (a research organization supporting FAA) has prototyped an AI tool that transcribes and analyzes ATC/pilot radio conversations to identify safety issues or training points [54]. NASA has developed speech recognition for ATC as part of projects like **SMART-**

NAS (researching how to incorporate voice assistants in control centers). These efforts contribute critical sub-components: high-accuracy transcription (dealing with noisy audio, various accents, and aviation jargon) and natural language understanding of controller commands. They are building blocks for an AI controller that can "hear" and "speak" on the radio.

- **Traffic Flow Management AI:** While our focus is on tactical control, AI is also being applied at the strategic level of traffic flow management. The earlier-mentioned *Flyways AI* by Airspace Intelligence is deployed with some users (reportedly including airlines or maybe FAA command center) to optimize routing around weather and congestion [55] [56]. It uses machine learning to predict future sector loads and recommend reroutes or flow restrictions proactively [57]. The significance is that AI is already trusted to influence ATM decisions on a broader scale (though final decisions still rest with humans). If an AI can gain confidence in making flow management suggestions that affect hundreds of flights (with demonstrated benefits), that bodes well for more localized AI decision support in ATC.

- **Digital Tower and Controller Assistance Products:** A few air navigation service providers (ANSPs) and companies have started implementing "digital tower" technologies. For example, *SAAB's Remote Tower* system (deployed in some European airports) uses high-definition cameras and can incorporate automated aids (like target tracking, alerting for runway incursions). While not LLM-based, it shows movement toward automation in the tower environment. Some systems include AI-based *"Intelligent Sequence Assistants"* (ISA) for approach sequencing. In an EU project called **HAIKU**, a prototype ISA (based on neural networks) was developed to help tower controllers optimize runway usage and spacing, providing real-time suggestions for arrivals sequence with more look-ahead than humans typically manage [58] [59]. Early results indicated improved efficiency and maintained safety, with the AI agent giving controllers additional decision-support in managing takeoffs and landings [59]. This kind of narrow AI – focusing just on one aspect like sequencing or spacing – could be integrated into a larger LLM-driven agent or used standalone at first.

- **Human-Autonomy Teaming Research:** Human factors experts have been actively studying how to insert AI into the controller's workflow without degrading performance. DLR (German Aerospace Center) recently presented a concept of a **"Digital Air Traffic Controller (ATCO) Assistant"** for en-route sectors [60]. Their approach explicitly designs new task distributions between the human and AI. For example, the digital ATCO might autonomously handle certain routine tasks (like altitude changes for cruise climbs or handoffs) while the human handles more complex decisions, or vice versa. They emphasize multiple modes: a human-only mode, a hybrid mode, and a fully autonomous mode where the AI runs the sector and the human just monitors [61] [62]. At Airspace World 2023, DLR demonstrated a proof-of-concept with an **integrated controller working position** featuring this AI assistant and a special interface for human-autonomy teaming. The interface provided transparency (e.g., highlighting flights where the AI proposed solutions, allowing the human to approve) and managed the communication flow between human, AI, and pilots. Such research offers practical insights into UI design and the concept of operations for mixed human-AI control. It acknowledges that simply dropping an AI into the system won't work; the procedures and tools must be adapted to facilitate *collaboration* between controllers and AI agents.

In summary, the current state of AI in ATM is one of **active experimentation and gradual progress**. We have components of the puzzle in place: speech recognition, conflict prediction, sequence optimization, even LLM-driven simulation of ATC dialogue. There is not yet a deployed AI that can assume a controller's full responsibilities, but each project expands the envelope. Importantly, these efforts also build a

knowledge base for certification – for instance, by studying an AI assistant in simulations, researchers can start to draft safety performance requirements and interface designs that would inform an eventual operational deployment. It's also clear that globally, many aviation stakeholders (NASA, FAA, SESAR, etc.) are **anticipating AI will play a growing role in ATC**, primarily as an aid to humans initially [63] [64]. The consensus is that early AI tools can make controllers' jobs easier and safer (e.g., by catching errors or suggesting optimal decisions), which in turn could help mitigate staffing issues and improve capacity.

Human Factors and Safety-Critical Design Considerations

Introducing an AI agent into ATC operations necessitates careful attention to human factors and safety-critical system design. Air traffic control is inherently a human-centered activity – even as automation increases, the *human controllers and pilots are ultimately responsible for safety*. Therefore, any AI assistant or automated controller must be designed as part of a **human-AI team**, not just a standalone system. Key considerations include:

Trust and Transparency: Controllers must be able to trust the AI's outputs, but not blindly. Achieving the right level of trust (neither distrust that wastes the AI's value, nor overtrust that could let an error slip through) is crucial. This hinges on transparency – the AI should be able to explain its reasoning in an accessible way. Features like the LLM's natural language rationale and the HAT (Human-Autonomy Teaming) interface showing what the AI is doing are intended to address this. For example, if the AI issues an unusual instruction, the system could display a note: *"AI rationale: vectoring behind landing traffic to maintain required spacing."* Early research shows controllers respond better to AI suggestions when they understand the why. Transparency also builds **accountability** – it should always be clear whether an action was taken by a human or machine, and on what basis, so that if something goes wrong, investigators can trace the decision process.

Workload and Attention: One might fear that adding an AI could *increase* a controller's mental workload (by having to monitor the AI in addition to everything else). The design must ensure the AI truly *reduces* net workload. This likely means the AI handles background tasks autonomously and only "surfaces" information or requests to the human when necessary. The interface should be *exception-based*: the AI deals with routine situations silently, but if it's unsure or if a decision is critical, it alerts the human. The human should not have to double-check every AI action – otherwise they lose any workload benefit. A good analogy is an auto-pilot in aircraft: pilots are aware of it but do not actively connstantly second-guess every minor control input it makes, they only step in when it kicks off or when parameters exceed normal. Similarly, controllers might come to treat the AI agent like an "auto-controller" handling mundane tasks in the background. A possible mode is that the AI manages 90% of standard instructions, and the human only issues or approves the 10% that are novel or exceptional.

Error Tolerance and Fail-Safe Design: In a safety-critical system, one designs for *graceful degradation*. The AI agent must be robust to errors in its input (e.g., a corrupted data feed or a misunderstood pilot transmission) and have strategies to fail safely. For instance, if the AI loses track of the situation or encounters something it doesn't know how to handle, it should default to handing control back to the human with an alert ("AI: Unable to resolve situation, human intervention required"). This is analogous to autopilots disengaging with an alarm when they reach the limit of their logic. Moreover, if the AI gives an incorrect instruction, the system should ideally have a multi-layer safety net: the verifier agent or rule-check catches it, but if not, the pilot or human controller should have the chance to recognize and correct it (which requires that transparency and training mentioned earlier). The **human must always have the final**

authority – a design principle likely mandated by regulators and ingrained in operating procedures. Controllers will need a straightforward way to override or veto the AI's actions at any time. This could be as simple as a physical "AI Hold" button that freezes AI clearances or a voice command like "stop" that the AI is programmed to heed immediately.

Training and Transition: Both controllers and pilots will require training to adapt to AI-augmented operations. Controllers will need to learn how to work with the AI tools – for example, interpreting AI-provided alerts, knowing when to rely on them versus manual judgment, and how to input any commands or corrections to the AI. This is a new skill set that training programs will have to incorporate (perhaps even new simulation scenarios where trainee controllers practice supervising an AI assistant). Pilots, on the other hand, might experience differences if AI is used for communications – e.g., more consistent phraseology, maybe faster responses. They might also need briefing on how an AI might handle certain situations (for instance, if an AI is controlling a sector and it issues an unexpected instruction, should the pilot treat it any differently? Ideally no, but these are considerations). The **rollout strategy** must ensure that at initial deployment sites, all personnel are comfortable and the human-AI team has been rehearsed in simulator scenarios including off-nominal conditions (e.g. how does the human step in if the AI malfunctions mid-rush hour?).

User Interface and Alerts: The design of the controller working position will likely change. Instead of just radar targets and flight strips, the interface might display AI suggestions (maybe as ghost targets or colored routes indicating the AI-proposed path). Alerts generated by the AI (e.g., conflict predictions) must be intuitively integrated – possibly an extension of current conflict alert systems but with reasoning attached. Care must be taken to avoid information overload; a cluttered screen of AI pop-ups would be counterproductive. Human factors research suggests using **ecological interface design** principles, where the AI's outputs are embedded in the same visual context controllers use, rather than separate windows [61] [65]. For example, if the AI plans to turn an aircraft, it could show a tentative turn arrow on the radar screen with a different color, which the controller can visually confirm or adjust before it's executed. If everything looks good, the controller might just let it happen (or click approve), and the arrow turns solid indicating an active clearance.

Performance and Bias: We must also consider how the AI's performance varies and ensure it doesn't introduce new types of errors. LLMs, for instance, can hallucinate – which in this context could mean giving a nonsensical instruction or mis-identifying an aircraft. Extensive domain fine-tuning and validation can mitigate this, but continuous performance monitoring is needed. The system should log all AI decisions and there should be a mechanism for post-event analysis. If an AI suggestion was inappropriate, developers need to analyze why (was it a gap in training data? A faulty prompt? etc.) and improve the system. There is also the matter of **bias**: training data might have inherent biases (for example, if most training scenarios were in clear weather, the AI might underperform in extreme weather). The development must consciously include diverse scenarios (thunderstorms, emergencies, heavy traffic, radio failures, etc.) in simulation training so the AI doesn't fail the first time it encounters one in reality. A safety-critical AI should undergo something akin to the thousands of hours of flight testing that a new aircraft autopilot does – exposed to every edge case possible to ensure reliability.

Psychological and Team Dynamics: Introducing AI will change the team dynamics in control rooms. There may be concerns among controllers about job security or changes in role. It will be important to frame the AI as a tool that *enhances* their job and safety, not a rival. In early deployments, keeping the human deeply involved (so they feel in control and responsible) will help acceptance. Over time, as trust builds, controllers

might willingly offload more routine tasks to the AI and focus on higher-level management. This parallels how pilots came to trust autoflight systems over decades – initial skepticism gave way to reliance after proving their worth, though even today pilots are trained to fly manually in case the automation fails. Controllers will likely always need to retain their core skills and certifications as a fallback.

Finally, from a **safety engineering** perspective, the AI system will need comprehensive hazard analyses (per FAA System Safety processes). Every potential failure mode (erroneous clearance, loss of network comms, corrupted input data, AI performance degradation, etc.) must have a mitigative design feature or procedure. For example, *Hazard: AI issues conflicting clearances to two aircraft*; *Mitigations:* conflict probe catches it and blocks the second clearance, auditory alarm to controller, controller immediately takes over communications. These kinds of scenarios must be considered and tested. The acceptable risk per FAA's Safety Management System is often quantified (e.g., no Category A runway incursions more frequent than 1 in 10 million ops, etc.); the AI system will need to demonstrate it can meet those targets with high confidence.

In conclusion, designing an AI ATC system is not just a technical exercise but a human factors one. **The human remains at the center** – at least until such distant future where full autonomy is proven – and the AI must be designed to seamlessly integrate into human workflows, providing tangible safety and workload benefits without undermining human authority or understanding. The system must tolerate errors gracefully and maintain the ultra-high safety standards of aviation. With careful design following these principles, the introduction of AI can be a net positive, potentially even *enhancing* safety by catching human errors (as many incidents are due to human mistakes under stress). But this will only hold true if the human-AI team is engineered and trained with the utmost diligence.

Implementation Roadmap and Risk Mitigation

Deploying an agentic LLM-based ATC system in live operations will require a **multi-phase implementation plan**, with each phase managing risks and feeding lessons into the next. Below is a plausible roadmap, aligned with FAA's incremental approach to AI and typical technology rollout practices in aviation [42] [66]:

Phase 0 – Development and Offline Testing:
Current state (Year 0-1): Build the prototype AI controller agent and test it extensively in offline environments. This involves training the LLM (including fine-tuning on ATC data), integrating it with a high-fidelity ATC simulation (e.g., NASA's ATM simulators or BlueSky), and iterating on the architecture. During this phase, the AI's performance on thousands of scripted scenarios is evaluated. We would create challenging traffic scenarios (including heavy traffic, emergencies, equipment failures) to see how the AI handles them. We also populate the experience library with diverse cases to improve its knowledge. Safety engineers conduct preliminary hazard assessments. At this stage, no real aircraft or live data – it's all simulation and perhaps shadow datasets from past days of traffic. The goal is to reach a point where the AI can **consistently manage a simulated sector without incidents** for extended periods and its decisions align with what an experienced human would do (or at least are safe if different).

Phase 1 – Shadow Mode in Live Environment:
Year 1-2: Deploy the AI system in a **shadow mode** at a real ATC facility. This means the AI receives live data feeds (radar, etc.) and listens to live communications, and it runs in parallel with human controllers, but **only as an observer**. It makes its own "decisions" and outputs (suggested clearances, alerts) that are recorded, but *it does not actually control anything*. The human controllers operate normally, unaware or not reliant on

the AI (or they might see AI suggestions on a separate screen experimentally, but they are instructed not to depend on them). This phase is about validation: comparing the AI's outputs to what the humans did. Did the AI spot the same conflicts and resolve them similarly? Did it perhaps catch something humans missed (that could be a near-miss if not caught)? Any discrepancies are analyzed. This shadow testing can run for months to build up statistical evidence of the AI's reliability. The FAA and dev team will particularly look at any "false alarms" or unsafe suggestions the AI made – these need to be ironed out. Shadow mode testing in aviation is common (it's used for new decision support tools, etc.) as it poses **zero risk** while providing a wealth of real-world data. By the end of this phase, we'd expect the AI to perform nearly identically to a controller in routine cases and to appropriately defer (i.e., "not sure what to do") in novel cases rather than act riskily.

Phase 2 – Human-in-the-Loop Trials (Limited Deployment):
Year 2-3: With confidence from shadow tests, begin controlled trials where the AI is allowed to actively participate, but with a human tightly in the loop. For example, at a less busy en-route sector or a low-traffic tower, introduce the AI assistant such that it can **propose clearances/actions to the human controller in real time**. The human has the final say – they either approve the AI's suggestion or modify it – then it is issued to aircraft. Essentially, the AI becomes a co-controller. The human is instructed to treat the AI like a trainee or junior partner: trust but verify. These trials might initially be done in simulators with certified controllers running traffic scenarios as if real, but quickly should move to live operations in low-risk periods (e.g., mid-night shifts with few aircraft, or managing spacing for arrival flows during good weather at a less complex airport). The goals here are to see how well the human and AI coordinate, to gather human feedback on the AI's usefulness or any UI issues, and to watch for any **failure modes** in an operational context. An important metric will be: does the AI actually reduce workload, or do controllers find it distracting initially? We also test contingency procedures: e.g., intentionally turning off the AI suddenly to ensure the human can take over seamlessly (the "uncoupling" scenario). If Phase 2 is successful, one would document that the AI never caused an unsafe situation, and ideally it prevented or improved handling of some situations. Regulatory observers (FAA safety reps) would be involved to evaluate if this can move to next step.

Phase 3 – Certified Operational Use (Incremental Rollout):
Year 3-5: After sufficient trials, the FAA may certify the AI assistant for operational use in specific contexts. This likely comes with lots of caveats and operating procedures spelled out. For instance, they might approve an AI-based **Conflict Advisory Tool** that is always running and providing alerts/recommendations to controllers, who are still in charge. This is akin to how TCAS was introduced on aircraft – first as an advisory that pilots could use at their discretion, later mandated because it proved its worth. The rollout would start at one or two facilities (say, one en-route center and one metro tower) as a beta deployment. Real controllers use the AI daily as part of their workflow. The **old system runs in parallel** for redundancy – e.g., controllers still have all their usual radar and strips, and perhaps even another person monitoring. As confidence builds, more facilities are equipped. Each deployment is monitored closely (through the Safety Management System process) to catch any site-specific issues. The FAA at this point would also finalize any rule changes necessary (for example, if the AI is to start directly issuing certain clearances, regulatory language might need to acknowledge automated instructions as valid). It's expected that *initial operational AI capabilities will be limited*: maybe just an automated clearance read-back and conflict warn at first, then gradually moving up to suggesting speed/heading changes, etc.

During this phase, there will likely be **continuous evaluation and iteration**. The AI models might be updated periodically as more data is gathered (with FAA approval). Controllers will report incidents or

glitches (like “AI gave an unnecessarily complex route for no reason”) which will be analyzed and corrected. The system might also expand functionality over time – for example, after proving conflict resolution suggestions work, they might enable an “AI spacing tool” for sequence optimization, etc. Each expansion is effectively a mini research-to-operation cycle requiring testing.

Phase 4 – Higher Autonomy and Scaling:
Year 5 and beyond: Assuming the AI has operated as a successful assistant and proven its safety, the door opens to attempt **higher levels of autonomy**. At this stage, one could trial a fully automated sector under supervision. For instance, an isolated piece of airspace at high altitude could be designated as an “AI-controlled sector” during certain times, with a human controller monitoring multiple such sectors remotely. The AI would issue clearances directly to pilots (with pilots aware via NOTAM that an automated controller is in operation, but procedures remain the same). The human monitor intervenes only if something seems amiss or the AI hands off control. This phase would be experimental and only after years of flawless lower-level operation. If it works, it could be gradually expanded, perhaps leading to a future where normal operations routinely involve AI managing less complex parts of the airspace or augmenting staffing during peak periods.

Throughout all phases, **safety risk management is continuous**. The FAA will likely require formal Safety Risk Management Documents (SRMDs) at each major change, analyzing hazards and mitigations, and only allowing progression if risk is acceptable and controlled. A guiding philosophy is that *the introduction of AI should never degrade safety – only improve it or maintain an already high level*. If at any point data suggests the AI is causing more issues than it solves, deployment would be paused or rolled back.

Key Risks and Mitigations:
It’s helpful to summarize major risks identified and how the plan addresses them:

- **Risk: AI decision error leading to conflict or accident.** *Mitigations:* Multi-layer safety checks (verifier agent, rule engine), human oversight in all but the final phased scenario, extensive testing on conflict scenarios, initial scope limited to advisory or non-critical decisions. Essentially, the AI is not given free rein until it’s statistically proven to be as safe as a top-notch human, and even then, a human or secondary system is watching.

- **Risk: Technical failure or downtime of AI system.** *Mitigations:* Gradual integration means that early on, if the AI fails, the human simply continues operation (no disruption). Later, redundant systems and a quick fallback to manual control are designed. The parallel operation of old and new systems (especially in Phase 3) ensures a fall-back. Also, robust engineering with redundancy reduces likelihood of a failure. The Newark blackout lesson [12] is heeded by requiring perhaps that the AI system undergo reliability testing and have backups (maybe an independent simpler conflict alert system always running as backup).

- **Risk: Controller overload or confusion when using AI.** *Mitigations:* Human factors design (HAT interface) and training. Phase 2 focuses on fine-tuning the UI from controller feedback. If controllers report confusion, adjustments are made (like simplifying how suggestions are presented). The system only adds to workload if not designed right, so that phase is critical to get the concept of operations correct.

- **Risk: Acceptance and organizational resistance.** *Mitigations:* Early and frequent engagement with controllers (through NATCA) and pilots (through ALPA and others). Show that the goal is to enhance safety and working conditions, not to eliminate jobs in a threatening way. The success stories (e.g., AI prevented an error or reduced delays) should be communicated to build support. Possibly, find champion users who become advocates when they see it works. Also, emphasizing that this helps cover staffing gaps and reduces overtime stress can make the workforce see it as a positive.

- **Risk: Regulatory delay or show-stopper.** The FAA could be very cautious and slow to approve anything. *Mitigations:* Work within their processes, involve their technical experts in the project from Day 1 (so they co-develop the safety requirements). Use the FAA Tech Center for testing, get buy-in by proving things in their view. Incremental demonstration of safety will be crucial – e.g., after shadow mode, present data to FAA that "AI would have prevented X number of pilot deviations or caught Y number of conflicts 2 minutes earlier than human" to show clear benefit. The phased approach itself is a mitigation, giving FAA multiple decision points to stop if unsafe.

- **Risk: AI cybersecurity threat.** If someone hacks or feeds malicious input to the AI, it could give dangerous instructions. *Mitigations:* Use closed networks (on-prem systems not internet-facing), rigorous cybersecurity certification (in line with FAA/NIST guidance [67]), and monitoring for anomalies in AI output that could indicate tampering. Possibly have the AI verify critical data with cryptographic signatures (for data from trusted sources only).

- **Risk: Unintended interactions with pilots (human factors).** Maybe pilots might not understand an AI controller's accent or pacing, or might be unnerved by a synthetic voice. *Mitigations:* Use high-quality speech synthesis with standard phrasing, maybe even pre-recorded human voice clips assembled to make it sound natural. Educate pilots about the system so they aren't surprised. Also ensure the AI follows standard phraseology 100% (no improvisation), which testing will ensure.

By Phase 4, if reached, the system would likely be robust and trusted enough that these risks are minimal. But the timeline can be adjusted based on findings – safety will dictate speed. If a major problem is found in Phase 2, one might spend extra years in development before moving on. It's also possible that the outcome of early phases is a decision that full autonomy isn't worth pursuing, but that the **assistant mode is extremely valuable** and gets widely deployed. Even that outcome addresses the core problem (controller shortage) by boosting productivity.

One encouraging thought is that similar roadmaps have been followed for past ATC enhancements. For instance, **Airport Surface Detection Equipment – Model X (ASDE-X)**, an AI-driven alert system for runway incursions, was introduced gradually and is now standard, providing automated warnings to controllers. Initially, controllers were wary of false alarms, but the system improved and is credited with preventing accidents. An AI controller can follow a similar path: initial skepticism, iterative improvement, eventual indispensable tool.

Conclusion

The vision of an AI-driven air traffic control system is no longer science fiction. With cutting-edge LLMs and AI agents demonstrating human-like reasoning and communication skills, we have the building blocks to augment – and in some cases, eventually automate – the roles of air traffic controllers. This could be transformational in addressing the controller workforce shortfall and scaling the aviation system to meet

future demand. Our analysis finds that **full automation of ATC, while potentially feasible in low-complexity scenarios, is a long-term goal** that must be approached incrementally. In the near term, the focus should be on **AI augmentation**: deploying LLM-based assistants that work side-by-side with controllers to enhance safety, reduce workload, and increase capacity. These AI agents can monitor vast amounts of data in real time, catch conflicts or errors before humans can, and suggest optimal solutions – essentially functioning as an ever-vigilant partner that never gets tired.

Technically, an agentic LLM architecture augmented with tools, memory, and rigorous constraints can encapsulate much of an expert controller's knowledge and decision process. The feasibility studies and prototypes cited (from SESAR's experiments to the Delft LLM conflict solver) provide confidence that such systems can be built and can perform at a high level in simulations [6] [52]. Real-world integration challenges like latency, data fusion, and reliable communications have known solutions within modern ATM infrastructure (e.g. SWIM, on-premise HPC deployments, proven speech recognition). The key hurdles lie in **validation and trust**: proving to regulators, controllers, and pilots that the AI will *enhance* the already exemplary safety record of air traffic control, not jeopardize it.

Regulatory acceptance will require exhaustive testing, transparency in design, and likely new standards for AI behavior and verification. Yet, the FAA's own plans and industry commentary indicate a recognition that AI is the "next leap" for ATC and a willingness to incorporate it prudently [4] [48]. By starting with decision support tools that pose minimal risk, and gradually increasing autonomy under strict oversight, the deployment can be done in a **risk-managed way**. Each step must demonstrably maintain or improve safety – for example, if AI reduces human errors or catches developing conflicts faster, those safety benefits should be quantified and fed back into expanding its role.

Our report also underscored the importance of **human factors** in this journey. The introduction of AI should relieve stress and workload for controllers, making the job more attractive and sustainable, which in turn helps solve staffing issues from another angle (better retention and recruitment) [11]. A well-designed human-AI teaming model will keep controllers in command, using the AI as a powerful tool – much like a skilled pilot uses an advanced autopilot. Pilots and the flying public may initially be wary, but as they experience consistent, possibly smoother ATC service (fewer holds, more optimal routings, timely conflict resolutions), confidence in the AI will grow. It's analogous to how we entrust autopilots for most of a flight or how many rely on AI-driven systems in cars – gradually, performance proves itself.

In closing, an agentic LLM-based ATC system is a **technically ambitious but achievable innovation** that directly targets a pressing need in the aviation industry. The path forward is one of evolution, not revolution: start small, validate each function of the AI in increasing complexity, and integrate it such that it complements human expertise. If successful, the outcome will be a hybrid human-AI air traffic management system that not only addresses the labor shortfall but also enhances the *safety, capacity, and efficiency* of U.S. airspace. Controllers will have an ever-ready assistant for the mundane tasks and an eagle-eyed second pair of eyes for the critical ones. Pilots will receive timely, data-driven guidance possibly improving flight efficiency. And passengers may experience fewer delays and a continued stellar safety record. The challenge is great – but so is the potential reward in modernizing one of the most critical human-operated systems of our time. With careful planning, robust engineering, and close collaboration between AI technologists, aviation experts, and regulators, the vision of AI-augmented or even AI-managed air traffic control can take flight.

Sources: The information in this report was drawn from a range of current research, industry white papers, and expert commentary. Notable sources include the FAA's AI safety roadmap [38] [39], academic studies on LLM-based ATC agents [5] [6], SESAR exploratory project results [35] [52], industry implementations like MosaicATM's voice analysis tools [19] and Airspace Intelligence's integration of FAA data feeds [15], as well as human factors analyses on human-AI teaming in aviation [68] [58]. These and other references are cited throughout the text to support the feasibility assessments, technical proposals, and regulatory/safety analyses presented.

---

[1] Raising ATC Retirement Age Not an 'Effective Solution' to Shortages, Union Says | FLYING Magazine
https://www.flyingmag.com/raising-atc-retirement-age-not-an-effective-solution-to-shortages-union-says/

[2] [8] [58] [59] [68] Human Factors Requirements for Human-AI Teaming in Aviation
https://www.mdpi.com/2673-7590/5/2/42

[3] [4] [11] [12] [37] [63] [64] [66] Comment: We need more air traffic controllers; they need AI tools | HeraldNet.com
https://www.heraldnet.com/opinion/comment-we-need-more-air-traffic-controllers-they-need-ai-tools/

[5] [10] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [49] [50] [51] [52] [53] Automatic Control With Human-Like Reasoning: Exploring Language Model Embodied Air Traffic Agents
https://arxiv.org/html/2409.09717v1

[6] arxiv.org
https://arxiv.org/pdf/2409.09717v1.pdf

[7] Why AI can't replace air traffic controllers - IMTS
https://www.imts.com/read/article-details/Why-AI-can-t-replace-air-traffic-controllers/1945/type/Read/1

[9] [19] [20] [21] Enhancing Air Traffic Communication Safety & Efficiency
https://mosaicatm.com/2025/04/02/enhancing-air-traffic-communication-safety-and-efficiency/

[13] System Wide Information Management (SWIM) - FAA
https://www.faa.gov/air_traffic/technology/swim

[14] The Next Generation Air Transportation System of the United States
https://www.sciencedirect.com/science/article/pii/S209580992100045X

[15] [16] [17] [18] [55] [56] [57] Air Traffic Management | ASI
https://www.airspace-intelligence.com/solutions/air-traffic-management

[38] [39] [40] [41] [42] [45] [46] [47] FAA Roadmap for Artificial Intelligence Safety Assurance, Version I
https://www.faa.gov/aircraft/air_cert/step/roadmap_for_AI_safety_assurance

[43] [44] [48] [67] In deploying AI, the Federal Aviation Administration faces unique challenges | FedScoop
https://fedscoop.com/in-deploying-ai-the-federal-aviation-administration-faces-unique-challenges/

[54] MITRE researched air traffic language AI tool for FAA, documents ...
https://fedscoop.com/mitre-air-traffic-conversation-ai-tool-faa-dot/

[60] [61] [62] [65] elib.dlr.de
https://elib.dlr.de/198901/1/Final_Enabling_Digital_Air_Traffic_Controller_Assistant_through_Human-Autonomy_Teaming_Design.pdf