Justin Cervero, Director of IT Systems & Research Computing  
Choya Durham, ITS Infrastructure Manager

Dear Justin and Choya,  
 On behalf of our CIS 4850 project team, we are pleased to submit our final report for the ATSD Ticket Automation project.  
 Over the course of the semester, our team analyzed the current ATSD call-handling process, designed an integrated solution using Cisco Webex Contact Center and Jira Service Management, and implemented a working prototype that automatically creates tickets for incoming calls. Our goal throughout the project was to reduce manual work for ATSD agents, improve data collection and accountability, and support a better service experience for students, faculty, and staff.  
 This report summarizes our business objectives, system requirements, architecture, development environment, and implementation details. It also includes a user manual for both end-users and administrators, as well as documentation of our project management process. This deliverable will be helpful not only as a record of our work but also as a foundation for any future enhancements and production rollouts.  
 Thank you very much for your time, feedback, and support throughout this project. We greatly appreciate the opportunity to work with ATSD on a real-world system that can meaningfully improve IT service delivery at Appalachian State University.  
Sincerely,  
The ATSD Ticket Automation Team  
- Patrick Masterson, Hank McAlister, Vedant Sheth, Matt Roehm

**ATSD Ticket Automation**

Final Report – Implementation and Delivery

Organization: Appalachian Technology Service Desk (ATSD)  
Course: CIS 4850 – Information Systems Project  
Professor / Project Supervisor: Dr. Jeff Kaleta  
  
Project Sponsor(s):  
 - Justin Cervero, Director of IT Systems & Research Computing  
 - Choya Durham, ITS Infrastructure Manager  
  
Team:  
 - Patrick Masterson – Project Manager  
 - Hank McAlister – Team Member  
 - Vedant Sheth – Team Member  
 - Matt Roehm – Team Member  
  
Date: December 2025

**Table of Contents**

[1. Introduction:](#_lo25ylz8nm4a) 4

[2. Business Strategy:](#_z0o7cmobjjqk) 5

[3. Project Definition - Team Charter:](#_15jvkhgsfmy6) 6

[4. System Requirements: 1](#_uu77srgjbifp)2

[5. Current Systems - Schematics and Description: 15](#_vqvaollr8z6d)

[6. Proposed System - Schematics and Description 17](#_p69omnc4rwk7)

[7. Development and Computing Environments 2](#)4

[8. Program Interface 2](#)5

[8.1 Graphical User Interface and Data Linkage 2](#)5

[8.2 Programming Components and Sample Code 2](#)6

[8.3 Middleware Functionality 2](#)8

[8.4 Outstanding System-Related Issues 2](#_m1umysci175q)9

[9. Value Added](#_cmqddb2f2mlf) 30

[10. Conclusion 3](#_ilqncou2dkrz)1

[10.1 Project Summary 3](#)1

[10.2 Meeting Client Requirements and Scope Changes 3](#)1

[10.3 Activities That Went Well 3](#)1

[10.4 Activities That Were Problematic 3](#)1

[10.5 Unexpected Occurrences 3](#_iy6vf73zyhwi)1

[11. References (APA) 3](#_1s7ioweyqko)2

[12. User Manual 3](#_8sheqilfza1)3

[12.1 End-User Manual (Caller and Agent) 3](#_xufoa2jej2zp)3

[12.1.1 Caller Experience 3](#)3

[12.1.2 Agent Experience 3](#)4

[12.2 Administrator Manual (System Admin / ATSD Lead) 3](#)6

[12.2.1 Key Components 3](#)6

[12.2.2 Business Hours and Closed Message 3](#)6

[12.2.3 Phone Number Collection Subflow 3](#)7

[12.2.4 Main Menu and Queues 3](#)8

[12.2.5 HTTP Request Configuration (Jira Ticket Creation) 3](#)9

[12.2.6 Screen Pop and Ticket ID Mapping 4](#)1

[12.2.7 Monitoring, Logs, and Troubleshooting 4](#_d1g77jk8wawz)2

[13. Team Management and Communication 4](#_dbhzuiwd4qwt)2

[15. Presentation Slides 4](#_4hoqm1aw1flg)4

**Table of Exhibits**

[Exhibit A: Current System: Context-Level Diagram 1](#_x5vj23ra6lna)6

[Exhibit B: Current System: Level-0 Logical Diagram 1](#_xuvtqb2bxnww)7

[Exhibit C: Proposed System: Context-Level Logical Diagram 1](#_3jk4yklcqgcg)9

[Exhibit D: Proposed System: Level-0 Logical Diagram](#_apozfkrx8sx0) 20

[Exhibit E: Proposed System: Context-Level Physical Diagram 2](#_im4grtgipldo)1

[Exhibit F: Proposed System: Level-0 Physical Diagram 2](#_q5kyt75t1pny)2

[Exhibit G: Entity Relationship Diagram 2](#_xyyeisw5rc0v)3

[Exhibit H: Webex Agent Desktop – Queue Login 2](#_vdmadc3lxjha)5

[Exhibit I: Jira Service Management – Auto-Created Ticket Example 2](#_m6e1rn5walgi)6

[Exhibit J: HTTPS Request Activity Configuration Output 2](#_e0rbjkylhqgx)8

[Exhibit K: JiraTicketPopUp Activity Configuration Output 2](#_ykv5eo8t4ulf)8

[Exhibit L: Middleware Variable Configuration 2](#_ncml21unuw5z)9

[Exhibit M: Sample IVR Menu Prompt 3](#_ibhfueszk0sy)4

[Exhibit N: Webex Agent Desktop Login 3](#_y414sscca1bw)4

[Exhibit O: Auto-Popped Jira Ticket 3](#_ww7pala1tprb)5

[Exhibit P: Business Hours Configuration 3](#_8mgx5v8nttp4)7

[Exhibit Q: PhoneNumberCollection Subflow Handoff 3](#_7brqa7vlk9nr)8

[Exhibit R: MainMenu IVR Configuration 3](#_ksqxhmdi5ewo)9

[Exhibit S: HTTP Request Configuration](#_8ah2rw9kjp83) 40

[Exhibit T: Jira Screen Pop Configuration 4](#_sijeunlow8ez)1

[Exhibit U: Project Work Breakdown Structure (WBS) 4](#_ehnsz4enjepj)2

# 1. Introduction:

The Appalachian Technology Support Desk (ATSD) serves as the primary help resource for students, faculty, and staff at Appalachian State University. The help desk receives continuous call volume related to password assistance, account access, AsULearn issues, technology troubleshooting, and general IT support. However, a recent internal audit revealed a critical gap in the ATSD workflow: there was no standardized process for tracking calls, documenting issues, or ensuring ticket consistency within Jira Service Management. This lack of documentation created major challenges, including reduced accountability, limited data for performance evaluation, and inefficiencies that hindered service quality.

Currently, ATSD operators rely on processes that are fully manual to gather caller information and create Jira tickets. Operators must verbally collect the caller’s phone number, issue category, and relevant notes, then manually input this information into Jira. This approach leads to inconsistent data entry, slower response times, and unnecessary administrative overhead. It also prevented the university from effectively analyzing call trends or making data-driven improvements.

The ATSD Ticket Automation System was developed to directly address these issues. The project integrates Cisco Webex Contact Center, Cloudflare Workers, and Jira Service Management into a seamless workflow that is automated workflow. Through Webex Flow Designer, the system can capture caller details, route calls appropriately, and automatically initiate ticket creation in Jira-dev using API-based integration. Instead of focusing on manual documentation, operators can prioritize assisting callers, improving efficiency, and user experience.

# 2. Business Strategy:

The ATSD Ticket Automation System directly supports Appalachian State University’s strategic goal of improving IT service efficiency, documentation accuracy, and operational accountability. By creating an automated workflow between Cisco Webex Contact Center and Jira Service Management, this project modernizes the help desk process while reducing reliance on manual effort.

Key Business Benefits:

1. Improved Efficiency:

* Operators no longer need to manually create tickets while assisting callers. Automation will reduce turnaround time and streamline call handling.

1. Enhanced Documentation Accuracy:

* Automated ticket creation ensures consistency. Caller phone numbers, issue categories, and operator assignments can be pre-filled and standardized.

1. Better User Experience:

* Callers experience faster routing, reduced wait times, and clearer options. Operators spend more time resolving issues and less time on administrative tasks.

1. Data-Driven Improvements:

* Ticket records enable reporting, analytics, training, and service optimization, capabilities previously impossible under the manual system.

1. Scalable Architecture:

* The integration uses cloud-based services (Webex, Cloudflare, Jira-dev) that can be extended with voice recognition, caller lookup, ticket categorization, and more advanced automation in future phases.

# 3. Project Definition - Team Charter:

This section includes the most recent and officially signed ATSD Ticket Automation Team Charter. The charter establishes the project purpose, scope, objectives, stakeholder information, requirements, risks, constraints, external dependencies, and formal sign-off from all team members, sponsors, and the project advisor. It guided the team’s roles, responsibilities, and collaborative processes throughout the duration of the project.

| General Project Information | |
| --- | --- |
| Project Name | ATSD Ticket Automation |
| Description | This project focuses on automating the Appalachian State University IT Help Desk workflow by integrating Cisco Webex Contact Center with Jira Service Management.  Currently, ATSD operators do not create any documentation/tickets from calls, and when they do, they must manually collect caller details and create tickets within Jira. This not only makes it impossible to track caller issues and record interactions, but it also slows response time, increases the chance of missing details, and adds friction for both users and staff.  The solution to this is to automate call intake, menu routing, and ticket creation through an intelligent voice-based flow that captures input and triggers API calls to Jira-dev, improving efficiency, accuracy, and user experience. The proposed system will enable automatic Jira ticket creation directly from Webex’s Flow Designer. This will streamline how IT support calls are logged and tracked. The integration of Webex Contact Center and Jira will capture caller data, operator information, and call category to generate pre-filled Jira tickets, reducing manual entry and response times. |

| Member Name | Role | Email | Telephone |
| --- | --- | --- | --- |
| Patrick Masterson | Project Manager | mastersonpf@appstate.edu | 936-232-5997 |
| Hank McAlister | Team Member | mcalisterhs@appstate.edu | 336-707-4273 |
| Vedant Sheth | Team Member | shethvh@appstate.edu | 828-803-2450 |
| Matt Roehm | Team Member | roehmmc@appstate.edu | 919-704-6190 |
| Justin Cervero | Sponsor | cerverojj@appstate.edu | N/A |
| Choya Durham | Sponsor |  | N/A |
| Dr. Jeff Kaleta | Advisor | kaletajp@appstate.edu | N/A |

| Stakeholders Information |
| --- |
| *The following is a list of all stakeholders who may have a particular interest in the ATSD Ticket Automation* *Project.*  *We believe this project will garner interest across several universities and technology teams, as a shared initiative to improve IT service efficiency, reduce operator workload, and enhance the end-user experience for students, faculty, and staff.*  *We hope to communicate closely with our stakeholders throughout development and testing phases to identify system requirements, mitigate business risks, and ensure the final solution meets operational expectations and user satisfaction goals.*  Primary Stakeholders:   * Appalachian State University IT Services (ATSD) – Beneficiaries of the automated help desk workflow. * Cisco Webex Contact Center Team (App State) – Provides Flow Designer and voice AI integration support. * Jira Development Team (AppState Jira-dev) – Receives ticket automation and workflow data. * End Users (Faculty, Staff, Students) – Callers benefiting from faster and more consistent IT support. * ATSD Ticket Automation Team (App State) – The developers creating this project (us).   Stakeholder Interests:   * AppState IT Services: Reduce response time and improve help desk efficiency. * Cisco / Jira Teams: Validate integration reliability and scalability. * University Administration: Improve user satisfaction and IT service reputation. |

| Project Scope Information |
| --- |
| Project Purpose / Business Justification |
| A prior audit recognized that the ATSD ticket process was not evaluated because a process for collecting data had not been established, and records had not been kept of previous tickets. This created a lack of accountability, a lack of records, and the inability to gauge service.  Currently, help desk operators rely on manual processes within Jira Service Management to document each caller’s information and issue details. In addition to a lack of data collection methods, this process potentially resulted in slower response times, inconsistent data entry, and increased workload for both staff and student operators. The project addresses the need for a centralized, reliable, and automated system where Appalachian State University’s IT Help Desk can efficiently manage incoming support calls and automate ticket creation while offering accountability of the process.  The ATSD Ticket Automation project creates a streamlined and intelligent communication link between Cisco Webex Contact Center and Jira-dev. By automating the ticket creation process through Webex Flow Designer and Cloudflare, operators can focus on assisting callers instead of manually generating tickets. The system will automatically log essential information such as the caller’s phone number, issue category, and assigned operator, ensuring accuracy, consistency, and faster turnaround times.  This integration fills a critical gap by reducing redundant administrative steps, enhancing the operator experience, and improving the overall efficiency of IT service delivery. It supports the university’s ongoing initiative to modernize its technology infrastructure and elevate the quality of user support for students, faculty, and staff.  The Webex–Jira integration will not replace the operator’s role, collect unnecessary personal information, or act as a decision-making system for IT issues. It will serve purely as an enhancement to the existing help desk workflow, improving efficiency, accuracy, and the end-user experience without adding complexity or administrative burden. |
| Objectives |
| *This project's design will deploy a working Webex Flow Designer call logic that:*   1. *Checks hours of operation.* 2. *Verifies or captures the caller’s phone number.* 3. *Presents a menu of call options (Password Assistance, AsULearn Support, Help Desk, Tech Support Center).* 4. *Routes calls appropriately to the ATSD operator queue.*   *This project will then integrate the Jira-dev API to auto-create a new ticket when a call is answered, pre-filling:*   * *Assignee: Operator’s username.* * *Reporter: Placeholder email (editable by operator).* * *Description: Caller’s issue and phone number.* * *Project Queue: technology-help-desk-thd.* * *Ensure secure data transfer with audit logging enabled.* |
| Requirements |
| *The following outlines the requirements for developing the Webex–Jira Help Desk Automation Integration, categorized into technical and user requirements. These specifications are designed to create a reliable, secure, and scalable automation framework that improves how Appalachian State University’s IT department manages incoming support calls. By integrating Cisco Webex Contact Center with Jira Service Management through Cloudflare, this project aims to establish a modern, voice-driven workflow that captures caller information, routes calls efficiently, and automatically generates support tickets. This system will enhance operational efficiency, reduce human error, and improve the overall user experience for both IT staff and the campus community, ensuring long-term maintainability, compliance, and expandability for future enhancements.*  Technical Requirements   * Platform: Cisco Webex Contact Center (Flow Designer) * Integration Layer: Cloudflare Worker (handles API routing & security) * Ticketing System: Jira Service Management (Jira-dev) * APIs Used: Jira REST API for ticket creation * AI Component: Webex Voice AI * Security: HTTPS, API tokens, Cloudflare Access policies * Testing Environment: Jira-dev queue and appstate custom agent desktop   User Requirements   * Operators must have a seamless call interface with automatic ticket creation. * Callers must be able to hear clear prompts, choose the right department, and provide details securely. * The system should allow fallback manual entry in case of integration failure. * Tickets should auto-open in a browser tab for operator follow-up. |
| Major Known Risks |
| In creating an automated integration between Cisco Webex Contact Center and Jira Service Management, there are several risks that could affect performance, reliability, or user experience. The following examples represent the most significant known risks, along with the strategies that will be used to mitigate them throughout development and testing. These risks primarily center around system compatibility, API reliability, and communication between platforms. Our goal is to identify and minimize these risks early in the process to ensure a smooth rollout and dependable functionality once the integration goes live.  **Webex–Jira API Misconfiguration → Medium**  An incorrect or incomplete API setup could prevent ticket creation or data transfer. This will be mitigated through sandbox testing and validation using mock calls before live deployment.  **Cloudflare Integration or Security Layer Failure → High**  Issues with Cloudflare routing or access control could interrupt communication between Webex and Jira, and there is very little our team can work around that, as we rely solely on Cloudflare to make this project work.  **TTS (Text-to-Speech) or Input Recognition Issues → Medium**  If the automated prompts or keypad inputs are unclear, users may select the wrong menu option or fail to provide the required information. To prevent this, TTS messages will be carefully scripted, tested, and refined for clarity and usability.  **Token Expiration or Authentication Failure → Medium**  Jira API tokens may expire or become invalid over time, halting ticket creation. This will be mitigated through periodic token rotation, alerting mechanisms, and documented procedures for renewal.  **Delays in Cisco Coordination or System Access → Medium**  Implementation timelines may be affected by Cisco approval cycles or AppState system access restrictions. To reduce this risk, regular check-ins and follow-ups have been scheduled in advance of key milestones with our sponsor.  **Operator Training and Change Management → Low**  Staff unfamiliar with the automated workflow may initially find the process confusing. This will be mitigated through demonstration sessions and providing clear guidance on fallback procedures if automation fails. |
| Constraints |
| The following constraints represent factors that may limit or affect the completion and functionality of the ATSD Ticket Automation Project. While the team has full access to Appalachian State University systems and development environments, there are still external and technical factors that must be considered and worked around during the implementation process.  **Cisco Licensing Limitations:**  The current Cisco Webex environment is dependent on available licensing features. Some advanced functionality, including extended API usage and third-party integration permissions, may require elevated licensing tiers or administrative approval from Cisco. This could temporarily limit the project’s ability to fully test or deploy certain flow components.  **Text-to-Speech (TTS) Trial Restrictions:**  The Webex Contact Center’s built-in TTS engine is currently operating under a trial configuration. This may result in limited runtime, restricted voice variations, or feature expiration during the development cycle. Testing will need to remain within these constraints until full production licensing or an alternative TTS engine is secured.  **API Rate Limits and Quotas:**  Jira’s REST API imposes rate limits that could restrict how frequently tickets are created during high call volume or stress testing. Proper throttling and queue management will need to be implemented to prevent request failures.  **Dependency on Cisco Cloud Services:**  Because the integration relies on Cisco’s cloud-hosted Flow Designer and telephony infrastructure, any outages, updates, or maintenance periods on Cisco’s end could temporarily impact functionality or delay testing.  **Ongoing Maintenance Requirements:**  As the integration depends on multiple interconnected platforms (Cisco Webex, Cloudflare, and Jira), each update or policy change may require periodic script and flow updates to maintain compatibility over time. |
| External Dependencies |
| *The following are the major external dependencies that this project relies on to function successfully. Each external service plays a key role in maintaining the automation workflow, ensuring security, and supporting day-to-day operations. Any disruption or limitation in these dependencies may directly impact the project’s performance or testing timeline.*  **Cisco Webex Cloud Infrastructure:**  Provides the telephony, routing, and TTS (Text-to-Speech) capabilities that form the foundation of the automated call logic. Webex’s uptime, licensing status, and system configuration are essential for testing and deployment.  **Cloudflare Services:**  Acts as the secure API gateway between Webex and Jira. Cloudflare handles SSL encryption, access control, and API forwarding logic, ensuring that all communications between systems are authenticated, secure, and monitored for potential threats.  **Jira-dev (Jira Service Management):**  The endpoint platform where help desk tickets are automatically generated and tracked. Jira-dev’s API stability, token authorization, and database availability are critical for ensuring that tickets are created accurately and without delay.  **Appalachian State University IT Network and Security Policies:**  The university’s internal IT infrastructure and security protocols govern network access, authentication policies, and integration permissions. Continuous alignment with AppState’s IT standards ensures that the automation remains compliant and sustainable long-term. |

| Sign Off |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name (printed)** | **Signature** | **Date** |
| Project Manager | Patrick Masterson | Patrick Masterson | 11/5/2025 |
| Team Member (1) | Hank McAlister | Hank McAlister | 11/5/2025 |
| Team Member (2) | Vedant Sheth |  | 11/5/2025 |
| Team Member (3) | Matt Roehm |  | 11/5/2025 |
| Sponsor | Justin Cervero |  | 11/6/2025 |
| Sponsor | Choya Durham | Choya Durham | 11/05/2025 |
| Course Instructor | Dr. Jeff Kaleta | (Signature obtained via email on November 12th, 2025) | 11/12/2025 |

# 

# 4. System Requirements:

The ATSD Ticket Automation System’s Requirements were defined early in the project lifecycle to ensure alignment with Appalachian State University’s IT needs, focusing on automation, efficiency, and scalability. These requirements are categorized into functional (must-have and nice-to-have), technical, and user requirements, drawing from stakeholder input, the project charter, and planning phases. They emphasize seamless integration between CISCO Webex Contact Center, Cloudflare Workers, and Jira Service Management while adhering to university security and privacy policies.

**Functional Requirements**

**Must Have (Core Scope):**

* **Call Flow Logic in Webex Flow Designer:** The system must check hours of operation, verify, or capture the caller’s phone number (via caller ID or manual input), present a clear menu of call options (e.g., Password Assistance, AsULearn Support, Help Desk, Tech Support Center), and route calls to the appropriate ATSD operator queue based on selection.
* **Automated Ticket Creation in Jira-dev:** Upon call answer, integrate with Jira's REST API to automatically create a new ticket in the "technology-help-desk-thd" project queue. Pre-fill essential fields, including:
* Assignee: Operator’s username.
* Reporter: Placeholder email (editable by the operator).
* Description: Caller’s issue details, phone number, and call category.
* **Error Handling and Fallback Mechanisms:** Include robust error handling for API failures, invalid inputs, or system downtime. Provide a fallback manual ticket creation workflow to ensure continuity if automation fails.
* **Data Integrity and Compliance:** Ensure accurate data capture and transfer without collecting unnecessary personal information. All processes must comply with university privacy policies, including secure logging of interactions for auditing.

**Nice-to-Have (Future Enhancements):**

* UI/UX improvements for operator interfaces, such as customizable dashboards or real-time call notifications.
* Advanced reporting and analytics for ticket trends, call volumes, and performance metrics.
* AI-driven features, like automatic issue categorization or voice recognition for enhanced input accuracy.
* Integration with additional tools, such as email notifications for ticket updates or multi-queue routing for expanded departments.

**Technical Requirements**

* Platforms and Integrations:
* Cisco Webex Contact Center (Flow Designer): Core platform for voice-based call routing, Text-to-Speech (TTS) prompts, and AI components. Must support intelligent voice flows for menu navigation and data capture.
* Cloudflare Worker: Acts as the secure integration layer for API routing between Webex and Jira. Handles HTTPS encryption, token-based authentication, access policies, and logging to ensure secure data transfer.
* Jira Service Management (Jira-dev): Ticketing endpoint using REST API for automated creation and updates. The environment includes a custom agent desktop for testing and production.
* Security and Protocols: All communications must use HTTPS, API tokens for authentication, and Cloudflare Access policies. Implement periodic token rotation, alerting for failures, and audit logging. No internet-exposed vulnerabilities; compatibility with university networks is required.
* Development and Testing: Scripting for auto-ticket generation with focus on payload accuracy, real-time updates, and scalability. Testing in sandbox environments (e.g., mock calls) before live deployment. Constraints include API rate limits, TTS trial restrictions, and dependency on Cisco licensing.
* Tools and Compatibility: No additional data collection beyond essentials. The system must be scalable for high call volumes and extensible for future phases, such as voice AI enhancements.

**User Requirements**

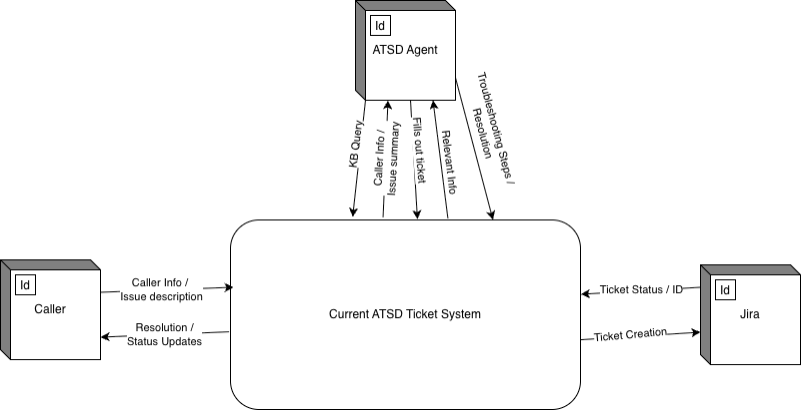
* Operator-Focused: Provide a seamless call interface with automatic ticket pop-up in a browser tab upon call acceptance. Minimize manual entry to allow focus on issue resolution. Include clear guidance for editing pre-filled tickets and fallback procedures.
* Caller-Focused: Deliver clear, audible TTS prompts for menu options and input requests. Ensure secure, efficient routing with reduced wait times and accurate issue logging.
* Administrator-Focused: Enable easy maintenance such as token renewal, script updates, and monitoring via logs. The system should support training with minimal disruption to live operations.

These requirements were validated through stakeholder consultations, including ATSD operators, Cisco and Jira teams, and university administration, to address the audit-identified gaps in documentation and efficiency. They form the foundation for the system's design, ensuring it meets immediate needs while allowing for future expansions.

# 5. Current Systems - Schematics and Description:

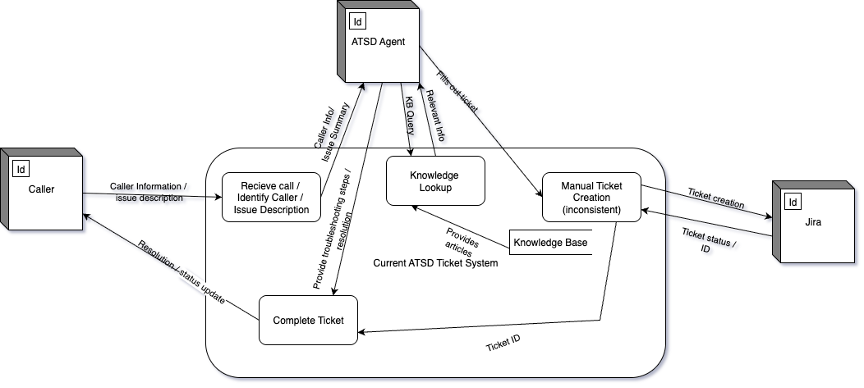
The current ATSD help desk workflow is fully manual and relies on operators to gather caller information, troubleshoot issues, and create Jira Service Management tickets by hand. When a caller contacts the university help desk, the request is routed through Cisco Webex and delivered to an ATSD operator using the App State Agent Desktop. The operator must verbally collect all relevant details, including caller identity, issue description, and category. They also have to manually type this information into Jira to create a support ticket. Overall, this whole process can be time-consuming and especially inconsistent across operators, which can make long-term reporting or performance tracking difficult.

### Exhibit A: Current System: Context-Level Diagram

******

The context-level diagram provides a high-level overview of the existing help desk workflow. It shows how the caller interacts with the ATSD operator through the Webex call system. The information, including the issue description, troubleshooting details, and ticket requirements, is exchanged verbally between the caller and the operator. The operator then manually enters this information into Jira Service Management, creating a ticket without any automated data transfer between systems.

### Exhibit B: Current System: Level-0 Logical Diagram

******

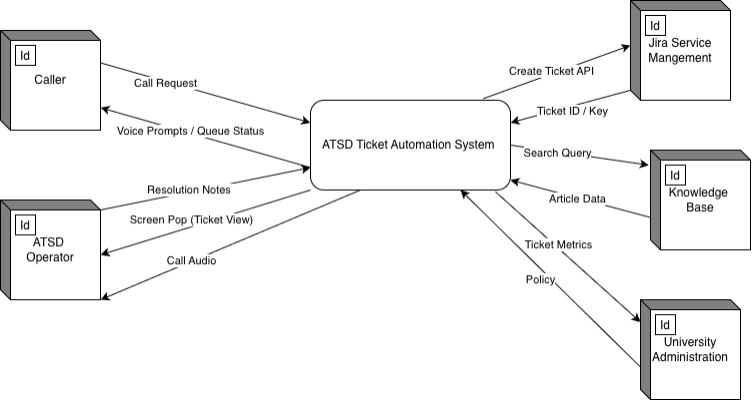
The Level-0 diagram expands on the manual processes performed by the ATSD operator. It details how the operator receives the incoming call, collects the necessary information from the caller, accesses internal resources such as the knowledge base, and manually inputs ticket data into Jira. This model highlights the many human-dependent steps in the workflow, such as issue clarification, information gathering, and ticket categorization. The absence of automation in these steps contributes to inconsistent documentation and slower service times

# 6. Proposed System - Schematics and Description

The proposed ATSD Ticket Automation System is designed to replace the existing manual ticket-creation workflow with an automated, integrated, efficient solution. Instead of relying on operators for information gathering when it comes to caller information and manually documenting tickets in Jira Service Management, the new system uses Cisco Webex Flow Designer, Cloudflare Workers, and Jira-dev to automatically generate a pre-filled help desk ticket as soon as an operator answers a call. This significantly improves automation, accuracy, and reduces operator workload. This provides stronger documentation and analytics capabilities for Appalachian State University IT Services.

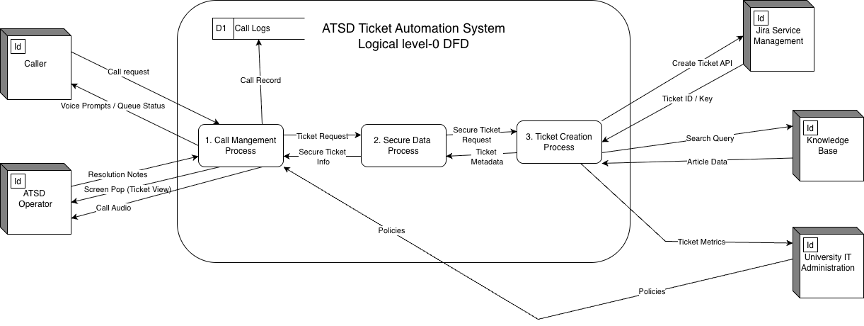
Our proposed system is designed using multiple layers of diagrams, including context-level logical, level 0 logical, context-level physical, level 0 physical, and our ERD diagram. Overall, our diagrams illustrate the high-level concepts and the detailed technical interactions of the new automated workflow.

### Exhibit C: Proposed System: Context-Level Logical Diagram

******

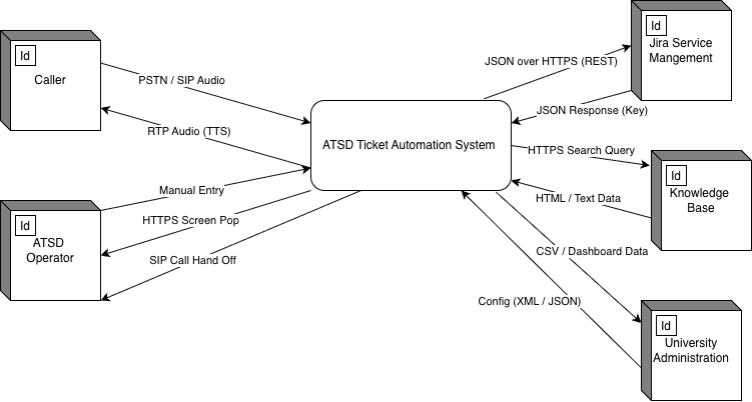
The context-level logical diagram illustrates a high-level overview of how the proposed automated system functions. In this design, the caller enters a Webex Contact Center IVR menu that captures their input and routes the call to an ATSD operator. At the same time, Webex generates structured call data and transmits it to a Cloudflare Worker, which formats and forwards the information to Jira Service Management via the Jira-dev API. Jira creates the support ticket and presents it to the operator, which eliminates the manual data entry process found in the current system.

### Exhibit D: Proposed System: Level-0 Logical Diagram

******

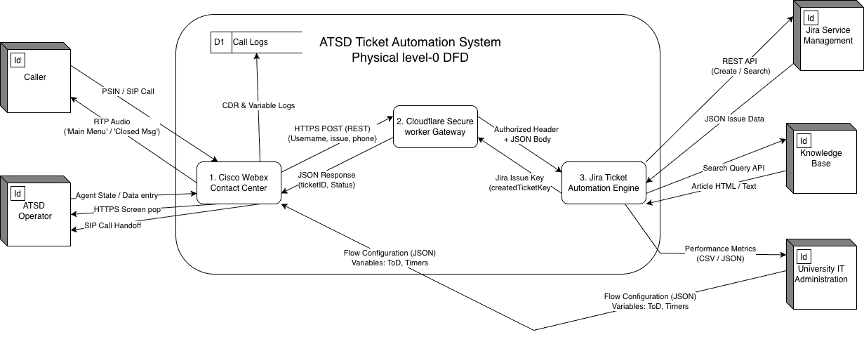
The level-0 logical diagram expands on the automated workflow by outlining each major process involved in ticket creation. Webex Flow Designer collects caller metadata, menu selections, and routing details, then sends a JSON payload to a Cloudflare Worker. The Worker parses the data, validates fields, formats the Jira request, and submits it to the Jira-dev REST API. Upon successful submission, a pre-filled ticket is automatically created and displayed to the ATSD operator, which ensures accurate and consistent documentation for every call.

### Exhibit E: Proposed System: Context-Level Physical Diagram

******

The context-level physical diagram illustrates the physical architecture of the proposed system. It highlights the integration between Cisco Webex Contact Center, Cloudflare Workers operating at the network edge, and Jira Service Management hosted by App State. All communication between systems occurs over secure HTTPS connections, and API authentication goes through Cloudflare access policies and Jira API tokens.

### Exhibit F: Proposed System: Level-0 Physical Diagram

******

The level-0 physical diagram presents a detailed view of the technical infrastructure powering the automation. It shows Webex’s Action Node generating a structured JSON payload, Cloudflare Workers performing serverless computation at the edge, and the Jira API endpoint receiving and processing ticket creation requests. The diagram also shows the secure HTTPS paths, authentication tokens, and system-to-system communication flows.

### Exhibit G: Entity Relationship Diagram

******

The ERD diagram provides a structural view of the data elements involved in the automated ticketing process. It illustrates how fields such as caller information, operator details, call metadata, ticket attributes, and category identifiers relate to one another within the Jira environment.

# 7. Development and Computing Environments

The solution was developed and tested in the following environments:

- Cisco Webex Contact Center (Flow Designer): Used to build the IVR menus, call routing, queue placement, and courtesy callback logic.

- Cloudflare: Used as a secure middleware layer for API forwarding, authentication, and logging.

- Jira Service Management (Jira-dev): Test environment used for ticket creation via REST API.

- Webex Agent Desktop: Browser-based interface where agents receive calls and see Jira tickets pop up.

Development followed an accelerated Work Breakdown Structure with planning, development, testing, deployment, and training phases from early November through early December.

# 8. Program Interface

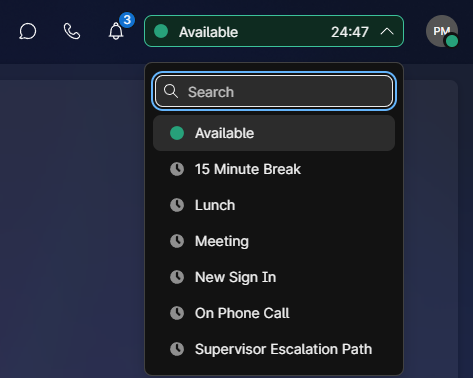
## 8.1 Graphical User Interface and Data Linkage

From the agent’s perspective, the key interfaces are:

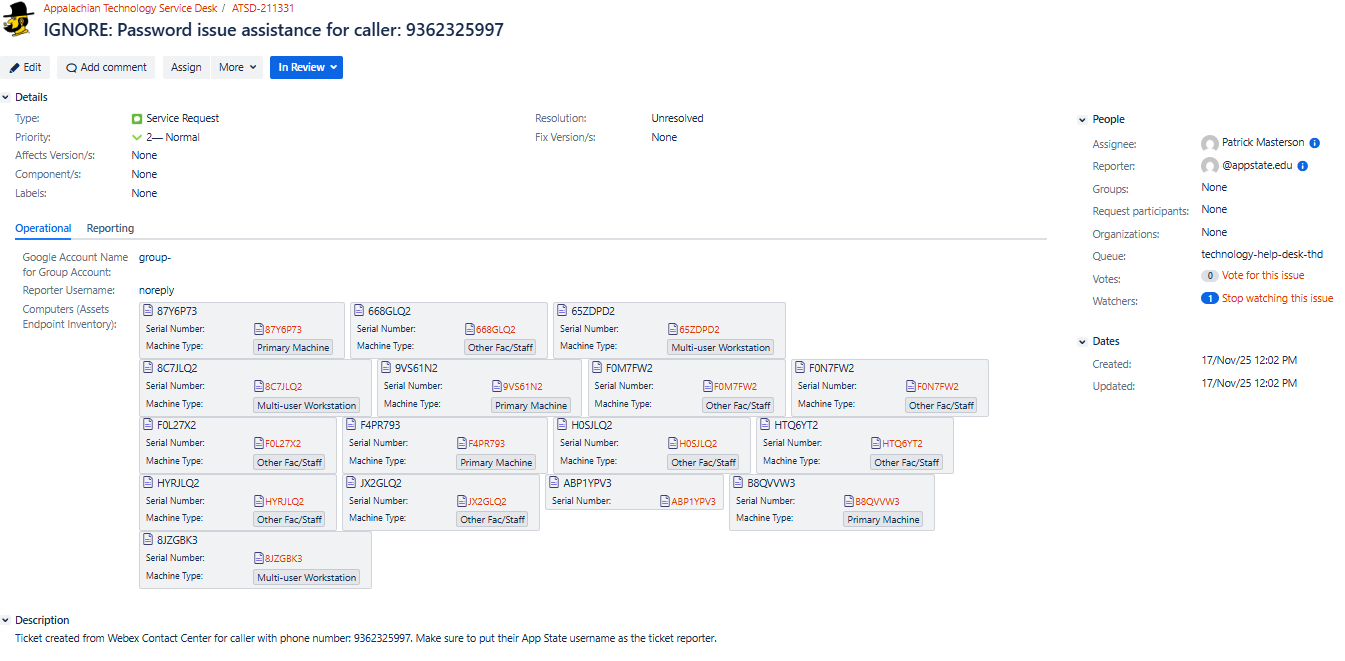
- Webex Agent Desktop (for logging in, joining queues, handling calls).

- Jira Service Management web interface (for working tickets that are automatically created and populated).

### Exhibit H: Webex Agent Desktop – Queue Login

****

### Exhibit I: Jira Service Management – Auto-Created Ticket Example

****

The Jira ticket is linked to the call via a summary line that combines issue type and caller phone number, and a description that indicates it was created from Webex Contact Center for the captured phone number.

## 8.2 Programming Components and Sample Code

The core automation is implemented in the Webex Flow Designer HTTP Request activity, combined with the Cloudflare/Jira integration.

Example HTTP Request configuration from the IT\_Service\_Desk flow:

- Method: POST

- URL: Jira REST endpoint or Cloudflare Worker URL.

- Headers: Authorization: Bearer {{HTTPRequestHeaderKey}}

Body (JSON):

{

"fields": {

"project": { "key": "ATSD" },

"issuetype": { "name": "Service Request" },

"summary": "{{issue}}: {{PhoneNumber}}",

"description": "Ticket created from Webex Contact Center for caller with phone number: {{PhoneNumber}}. Make sure to put their App State username as the ticket reporter.",

"reporter": { "name": "@appstate.edu" },

"assignee": { "name": "{{AgentAnswered.AgentEmailID}}" },

"customfield\_11607": { "name": "technology-help-desk-thd" },

"customfield\_12308": { "name": "technology-help-desk-thd" }

}

}

The response is parsed into the ticketID variable using JSONPath expressions such as $.id and $.key, which is then used by the JiraTicketPopUp activity to open the ticket in the agent’s browser.

### *Exhibit J: HTTPS Request Activity Configuration Output*

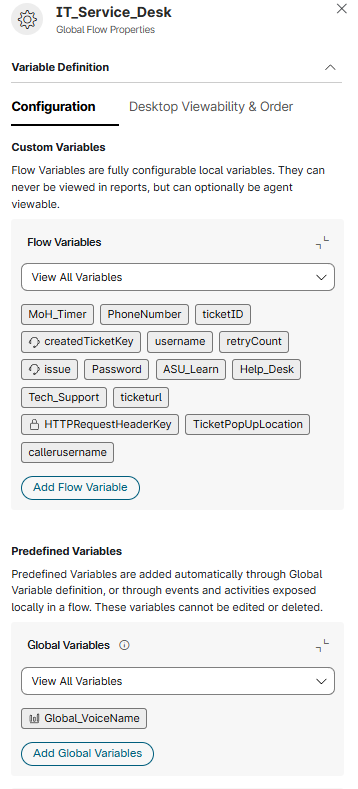
### *Exhibit K: JiraTicketPopUp Activity Configuration Output*

## 8.3 Middleware Functionality

Cloudflare Workers forms the middleware layer between Webex and Jira by validating incoming requests and tokens, forwarding payloads to Jira REST API over HTTPS, and optionally performing logging and minimal transformation of fields.

In the Webex flow, middleware-related behavior is supported by global and local variables (for example, HTTPRequestHeaderKey, ticketID, TicketPopUpLocation) and activities such as the HTTPRequest and JiraTicketPopUp activities.

### Exhibit L: Middleware Variable Configuration

****

## **8.4 Outstanding System-Related Issues**

Known or potential issues include:

- Webex–Jira API misconfiguration or credential expiration may temporarily prevent ticket creation.

- Cloudflare integration failures may impact communication between Webex and Jira.

- Text-to-speech or input recognition problems could affect user experience in IVR menus.

- Operator training and adoption must be monitored to ensure consistent use of automated tickets.

These issues can be mitigated through monitoring, validation before production changes, and clear troubleshooting procedures in the Administrator Manual.

# **9. Value Added**

The ATSD Ticket Automation project delivers the following value:

1. Improved Accountability and Analytics – Every answered call generates a Jira ticket, enabling metrics such as call volume, issue categories, and resolution times.

2. Reduced Manual Workload – Operators no longer need to create tickets manually while callers are on the line, which reduces errors and speeds up service.

3. Enhanced User Experience – Callers receive more consistent service and faster response times while operators focus on resolving issues rather than data entry.

4. Future Enhancement Potential – The architecture supports future additions such as advanced reporting, multi-channel integration, and AI-based routing or categorization.

# 

# **10. Conclusion**

## **10.1 Project Summary**

The ATSD Ticket Automation project successfully integrated Cisco Webex Contact Center with Jira Service Management to automatically create tickets for incoming calls. The solution addresses a long-standing audit gap and improves accountability, operator efficiency, and user experience.

## **10.2 Meeting Client Requirements and Scope Changes**

The project met core client requirements for automated ticket creation, time-of-day handling, menu routing, and queueing, adapted to a compressed timeline with an accelerated Work Breakdown Structure, and pivoted from an initial BarHub mobile app concept to a more focused, high-impact ATSD automation project when the original scope proved too large.

## **10.3 Activities That Went Well**

Strong collaboration with sponsors and advisors helped refine the scope and requirements, and the effective use of Webex Flow Designer, Cloudflare Workers, and Jira REST API delivered an end-to-end integration.

## **10.4** Problematic Activities

External constraints such as licensing, access to environments, and third-party reliability created schedule risk. The compressed project timeline required careful prioritization and deferral of some nice-to-have features to future phases.

## **10.5 Unexpected Occurrences**

Sponsor enthusiasm for the potential production rollout and future expansion (training, analytics, AI routing) exceeded expectations and highlighted the real-world impact of the project.

# **11. References (APA)**

Cisco Systems. (2024). Cisco Webex Contact Center – Administration guide. Cisco Systems.

Atlassian. (2024). Jira Service Management Cloud REST API reference. Atlassian.

Cloudflare. (2024). Cloudflare Workers developer documentation. Cloudflare.

Appalachian State University – ATSD Ticket Automation Team. (2025). Project Charter – ATSD Ticket Automation. Appalachian State University.

Appalachian State University – ATSD Ticket Automation Team. (2025). Planning and Requirements – ATSD Ticket Automation. Appalachian State University.

# 

# **12. User Manual**

## 12.1 End-User Manual (Caller and Agent)

### **12.1.1 Caller Experience**

1. Dialing the ATSD Number

- The caller dials the ATSD main help desk number.

- The system announces whether ATSD is open or closed based on business hours.

2. Phone Number Confirmation

- If the caller’s number is available from caller ID, the system confirms it.

- If not, the system prompts the caller to enter their phone number using the keypad.

3. Menu Selection

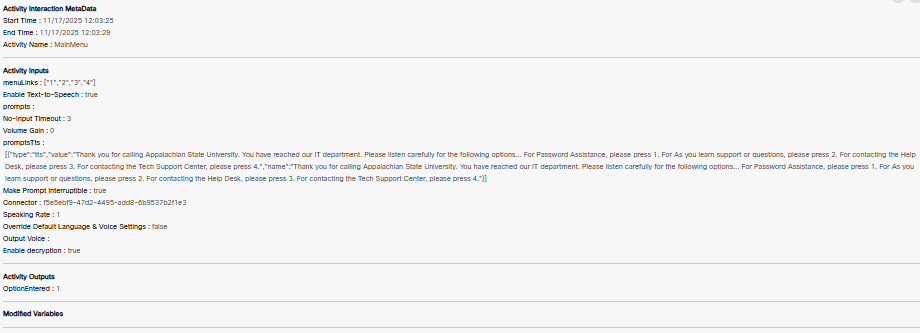
- The caller hears options such as:  
 - Press 1 for Password Assistance  
 - Press 2 for AsULearn  
 - Press 3 for Help Desk  
 - Press 4 for Tech Support Center

4. Queue and Hold

- The caller is placed into the appropriate queue with music on hold.

- A courtesy callback option may be offered so the caller can retain their place without waiting on the line.

### Exhibit M: Sample IVR Menu Prompt

****

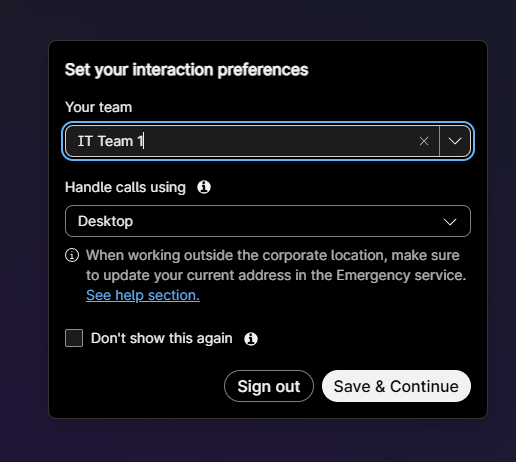
### **12.1.2 Agent Experience**

1. Logging into Webex Agent Desktop

- Agents open the Webex Agent Desktop in a supported browser.

- They log in with their App State credentials and join the appropriate queues (Password, AsULearn, Help Desk, Tech Support).

### Exhibit N: Webex Agent Desktop Login

****

2. Receiving a Call and Ticket Auto-Creation

- When a call is delivered, the agent answers as usual.

- The Webex flow automatically triggers the HTTP request, which results in a Jira ticket being created.

3. Jira Ticket Screen Pop

- The ticket ID returned from Jira is used to pop the ticket in the agent’s browser using the JiraTicketPopUp activity.

- The ticket summary includes the caller’s phone number and a brief description tied to the menu option.

### *Exhibit O: Auto-Popped Jira Ticket*

4. Working the Ticket

- The agent confirms the caller’s App State username and updates the Reporter field.

- The agent adds notes to the Description field, sets the appropriate priority, and follows standard Jira workflows for resolution.

5. Closing the Ticket

- Once the issue is resolved or transferred, the agent updates the ticket status according to ATSD policy.

## **12.2 Administrator Manual (System Admin / ATSD Lead)**

This section is based on the IT\_Service\_Desk Webex flow and the documented architecture.

### **12.2.1 Key Components**

- Webex Flow: IT\_Service\_Desk in Webex Flow Designer.

- Variables:

- PhoneNumber – Stores caller's phone number.

- issue – Stores a short description tied to the menu choice (Password, AsULearn, Help Desk, Tech Support).

- ticketID – Stores Jira ticket ID/key returned by the HTTP Request.

- HTTPRequestHeaderKey – Stores the Jira API token (secure).

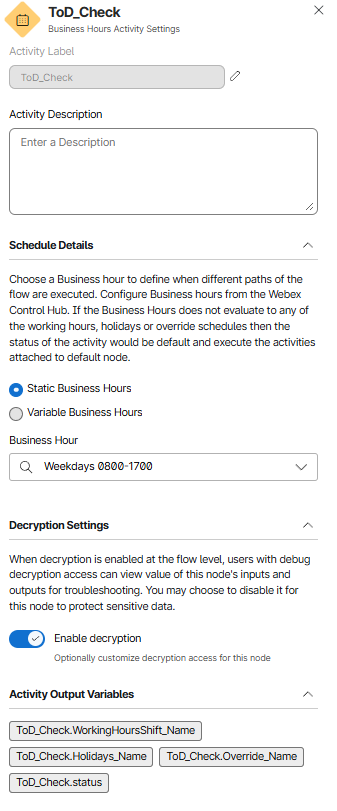
- TicketPopUpLocation – Base URL for Jira ticket screen pop.

### **12.2.2 Business Hours and Closed Message**

1. In Webex Flow Designer, open the Time of Day (business hours) activity and confirm that hours are correctly set for the agreed operational window.

2. Ensure the closed message activity contains the correct after-hours prompt and routes appropriately.

### Exhibit P: Business Hours Configuration

****

### **12.2.3 Phone Number Collection Subflow**

1. Validate that the PhoneNumberCollection subflow is configured and published.  
2. Confirm mapping of the PhoneNumber variable in both input and output parameters of the PhoneNumberCollection activity.

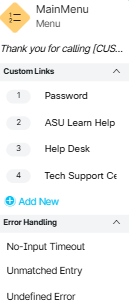
### *Exhibit Q: PhoneNumberCollection Subflow Handoff*

### **12.2.4 Main Menu and Queues**

1. Edit the MainMenu IVR activity and ensure that menu options map to the correct queue activities:  
 - Key 1 → Password Assistance queue  
 - Key 2 → AsULearn queue  
 - Key 3 → Help Desk queue  
 - Key 4 → Tech Support Center queue

2. Confirm each queue activity points to the correct Webex queue ID configured by ATSD.

### Exhibit R: MainMenu IVR Configuration

****

### **12.2.5 HTTP Request Configuration (Jira Ticket Creation)**

1. Open the HTTP Request activity used for Jira ticket creation.  
2. Verify that:  
 - Method is POST.  
 - URL is the Jira-dev REST endpoint or Cloudflare Worker endpoint.  
 - Authorization header uses Bearer {{HTTPRequestHeaderKey}}.  
 - Request body matches the field mapping required by Jira.  
3. Update HTTPRequestHeaderKey with a new API token whenever credentials change, and restrict access to admins.

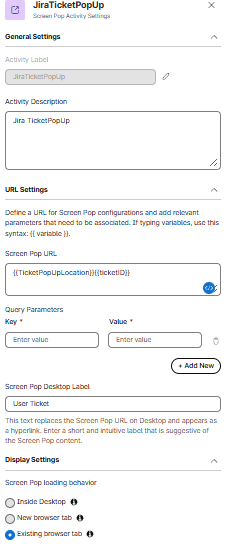
### Exhibit S: HTTP Request Configuration

****

### **12.2.6 Screen Pop and Ticket ID Mapping**

1. Confirm that ticketID is mapped correctly in the HTTP Request output using JSONPath.  
2. In the JiraTicketPopUp activity, ensure the screenPopUrl is set to something like:  
 {{TicketPopUpLocation}}{{ticketID}}  
3. Validate that the target is set appropriately (for example, same browser tab or a new tab) based on ATSD preference.

### Exhibit T: Jira Screen Pop Configuration

****

### **12.2.7 Monitoring, Logs, and Troubleshooting**

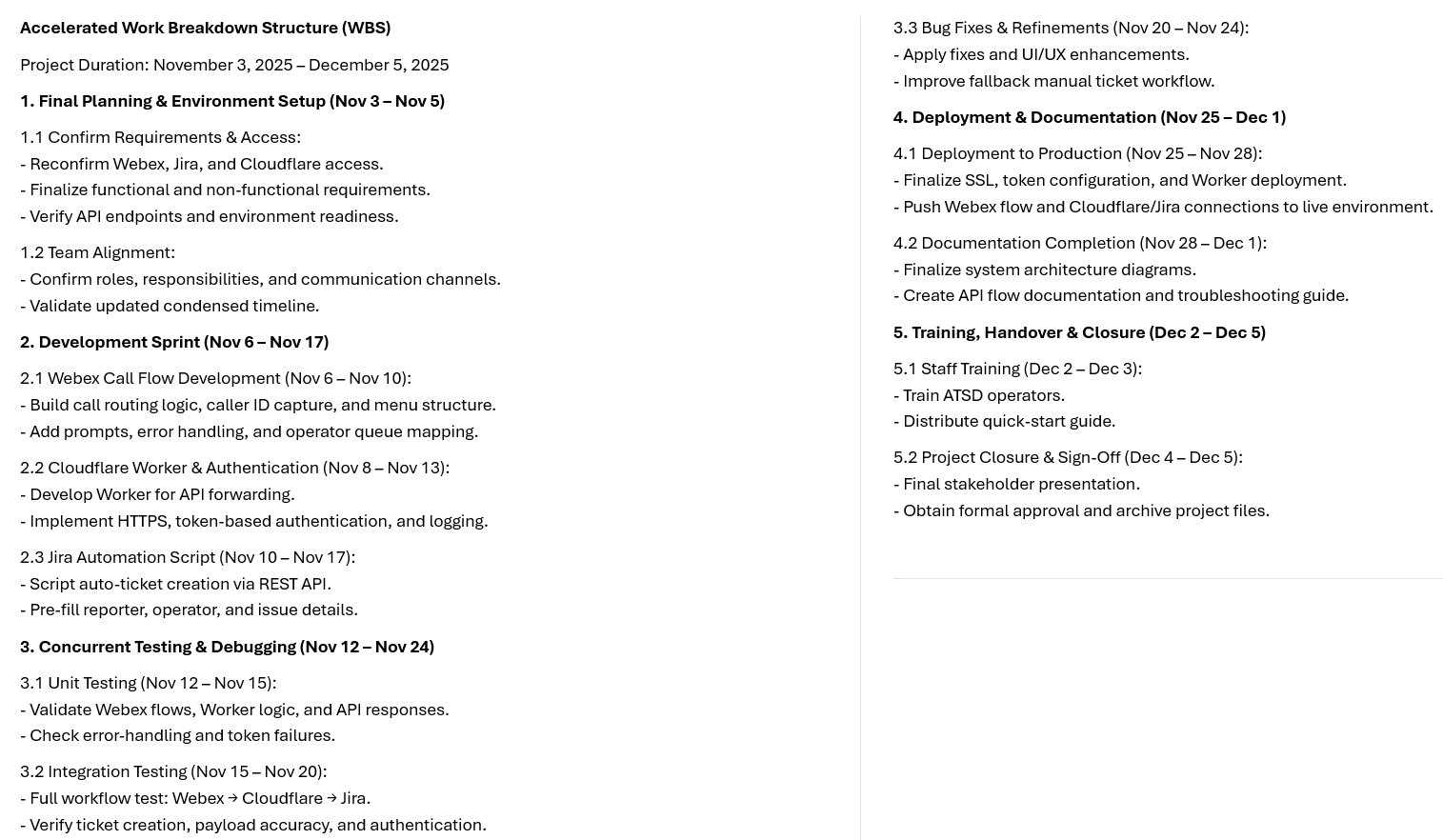
- Use Webex Flow diagnostics to review failed HTTP requests or misrouted calls.  
- Review Cloudflare Workers logs for API call failures, authentication issues, or timeouts.  
- Monitor Jira to ensure tickets are being created and to review any error responses.

If tickets fail to create, instruct agents to revert to manual ticket creation in Jira, check for API changes or expired tokens, and test the HTTP Request activity with a sample payload to confirm connectivity and correct field mapping.

# **13. Team Management and Communication**

Project management relied on an Accelerated Work Breakdown Structure covering planning, development, testing, deployment, and training.

### Exhibit U: Project Work Breakdown Structure (WBS)

****

Team communication included regular meetings with sponsors for feedback and alignment, weekly internal check-ins to track progress and issues, and milestone demos to show functionality and gather input.

Success factors:

- Early pivot from BarHub concept to ATSD automation, aligning scope with capabilities and timeline.

- Strong sponsor engagement and continuous feedback.

Lessons learned:

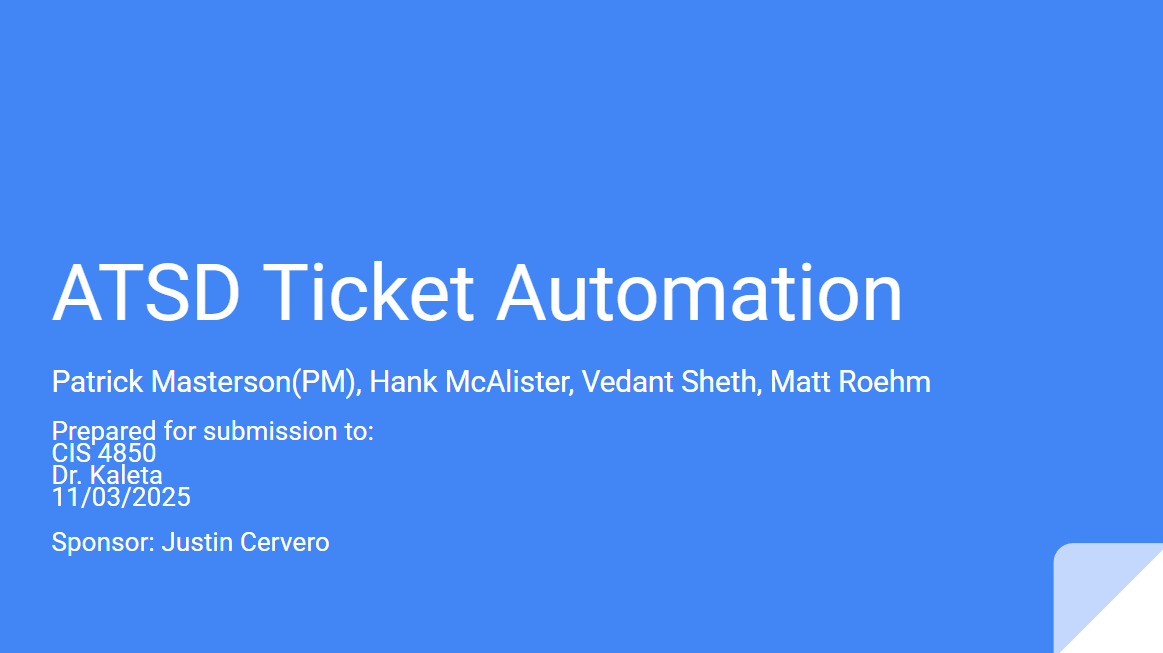
- Scope control and risk management are critical with third-party platforms and constrained calendars.

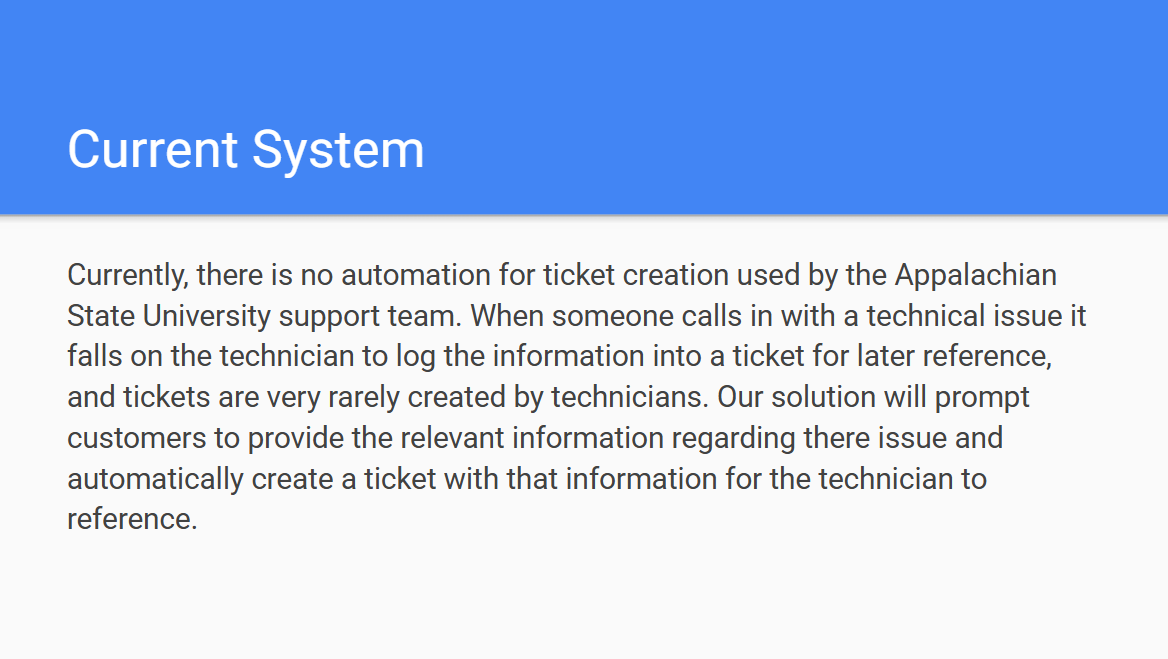
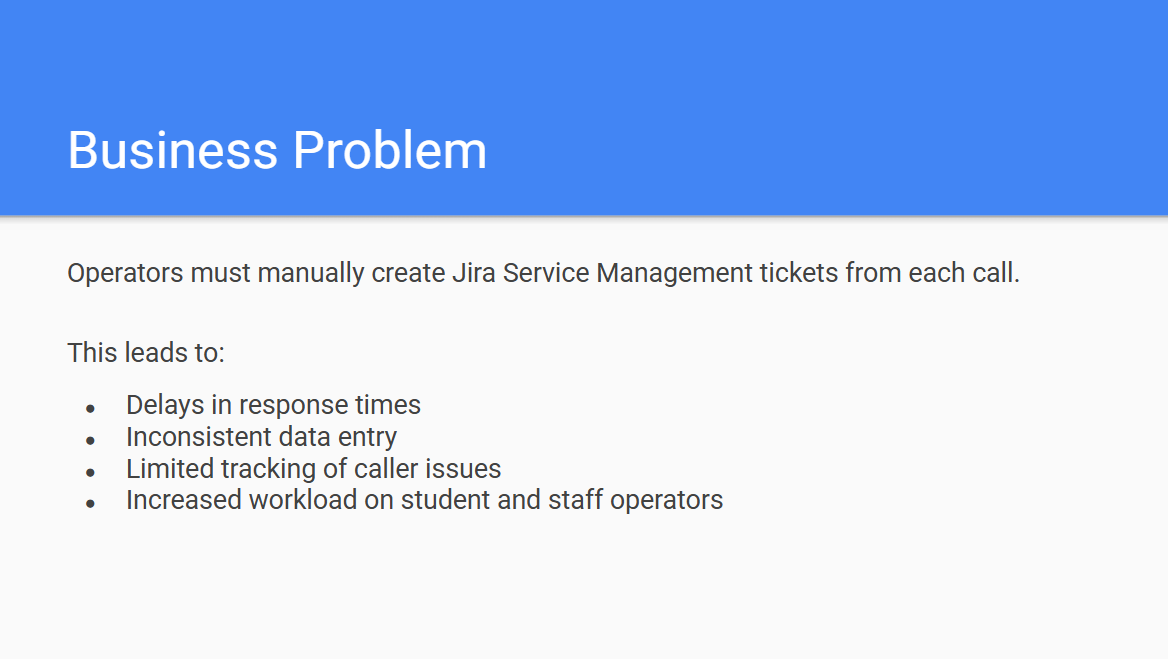
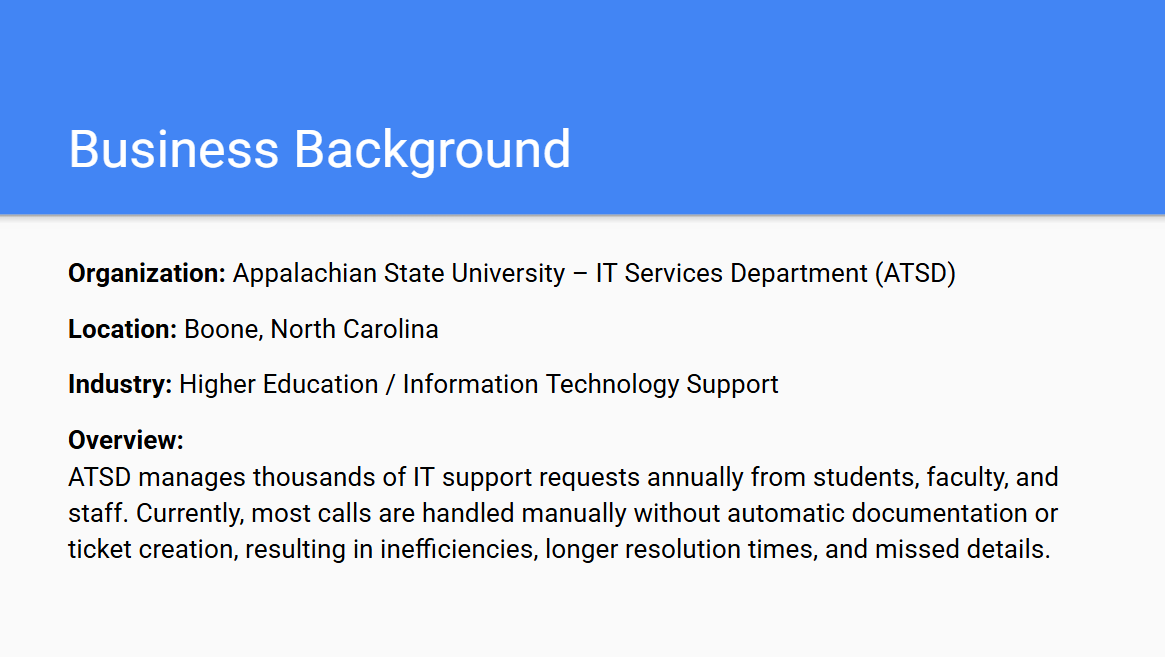
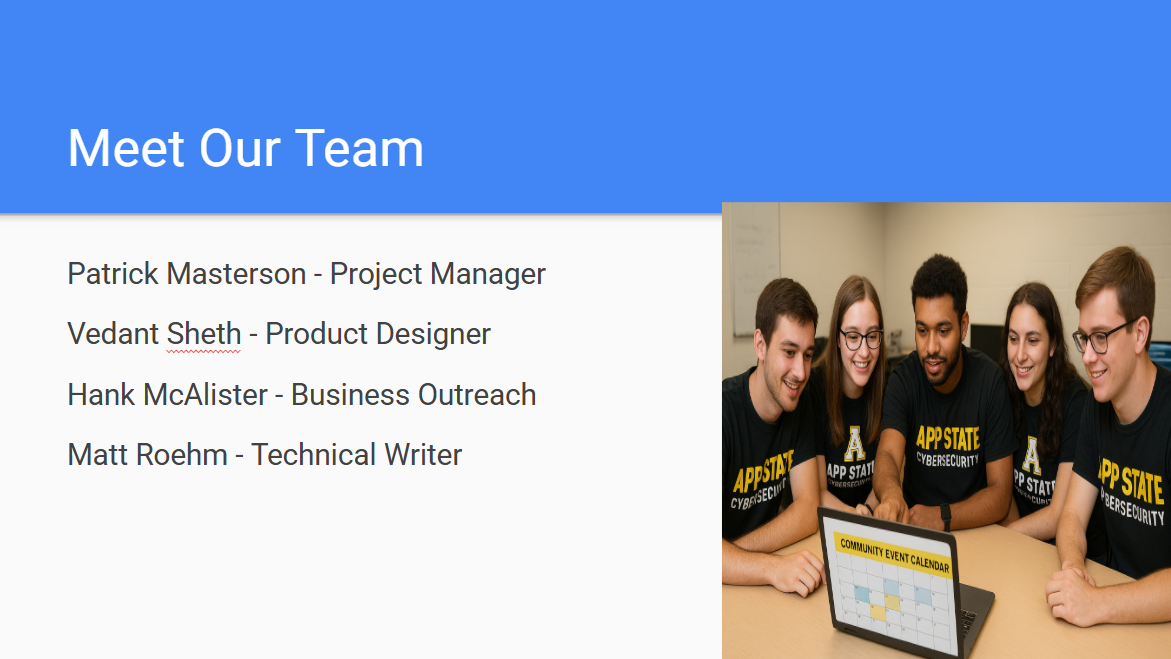
- Clear division of responsibilities within the team improves execution speed and reduces confusion.

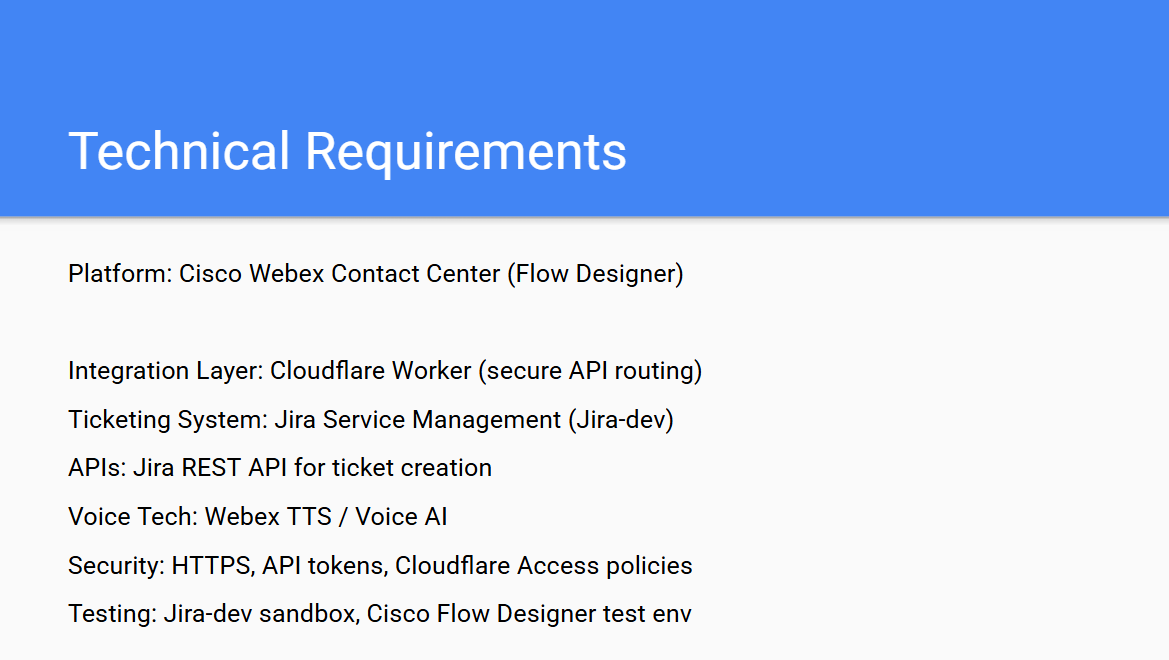
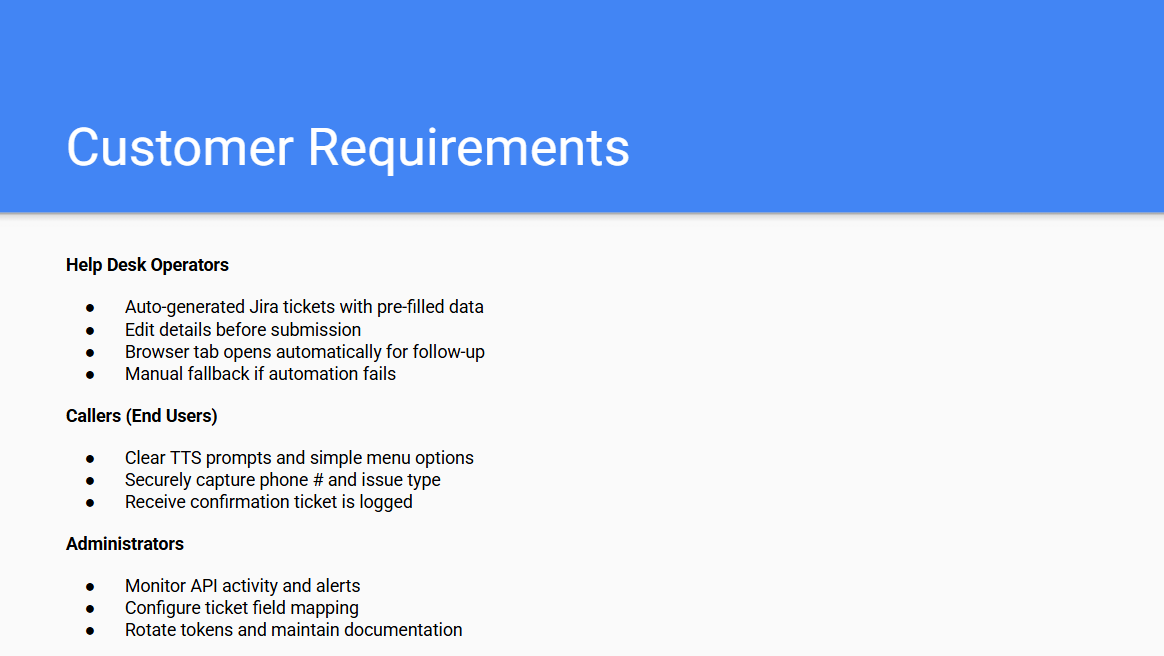
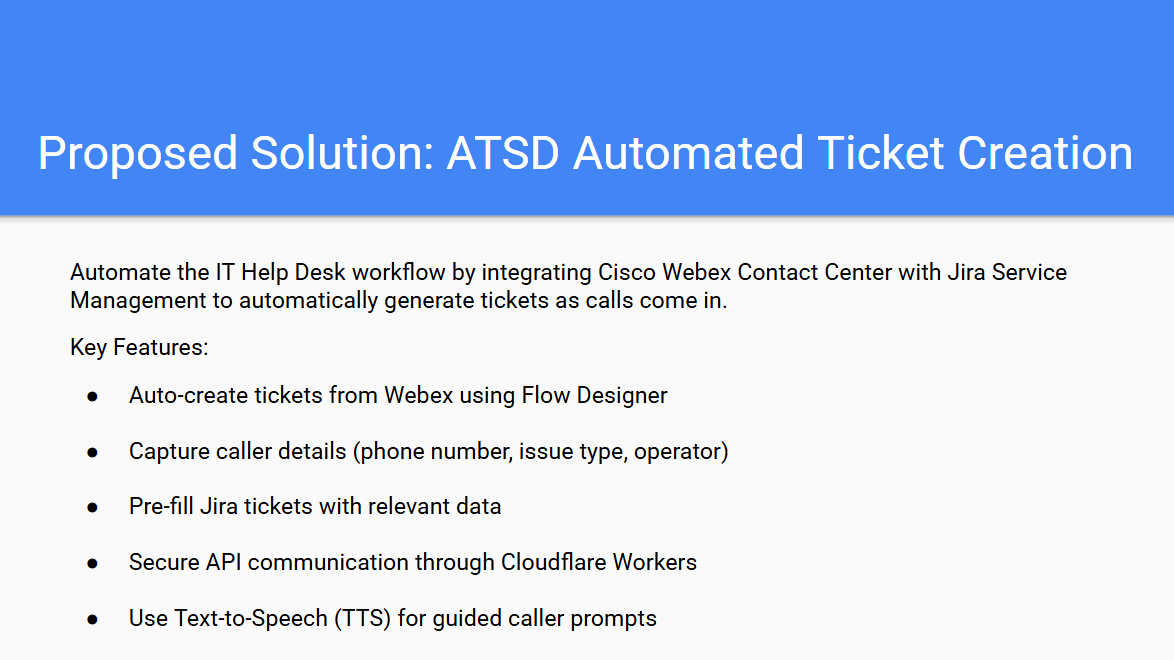
# 

# **15. Presentation Slides**

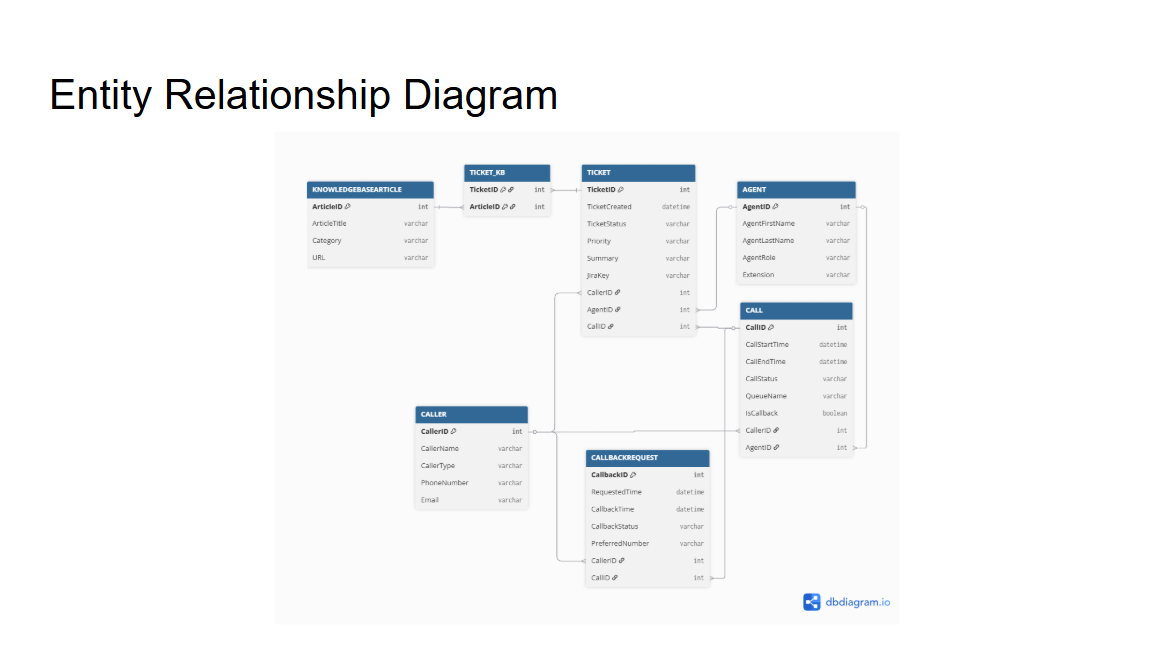
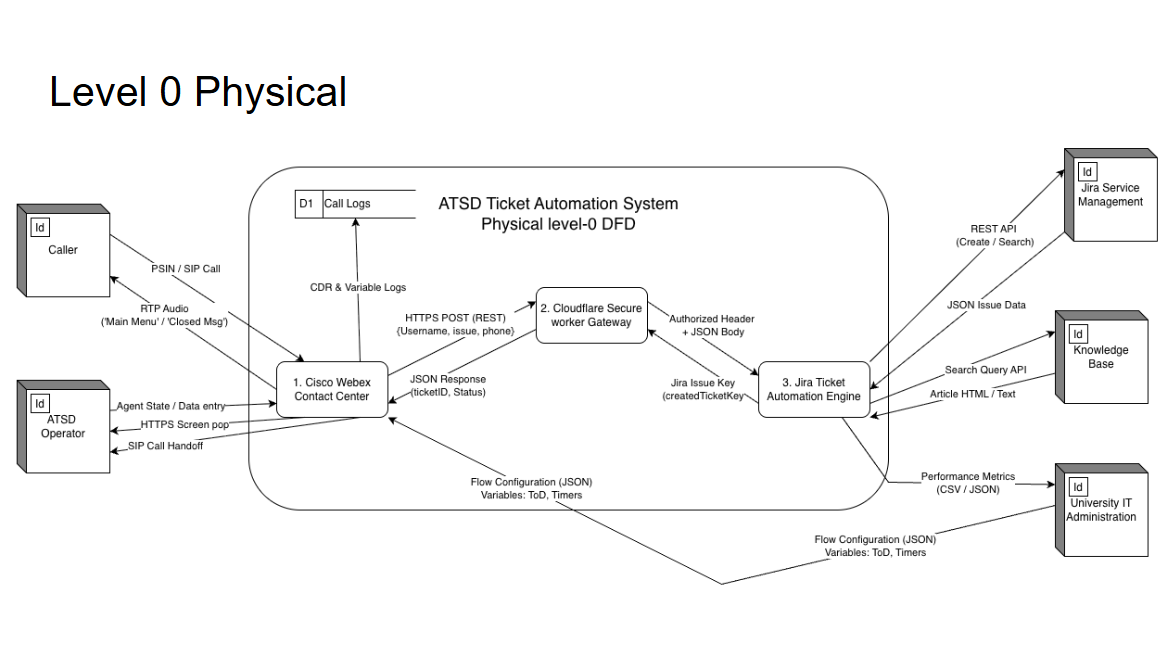
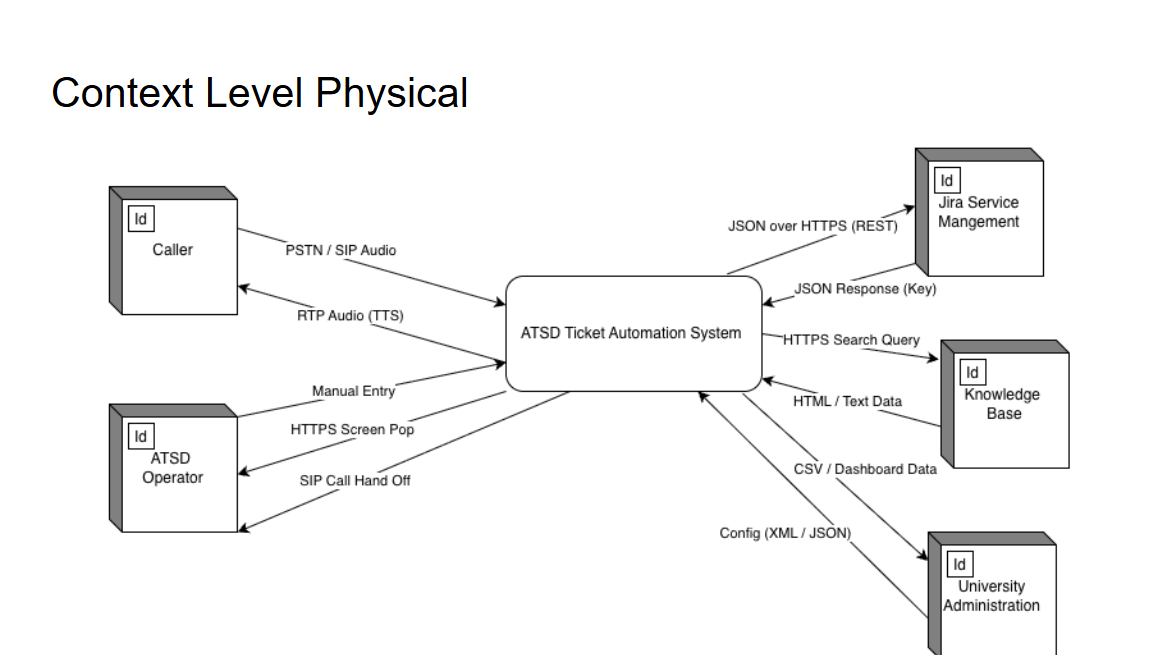
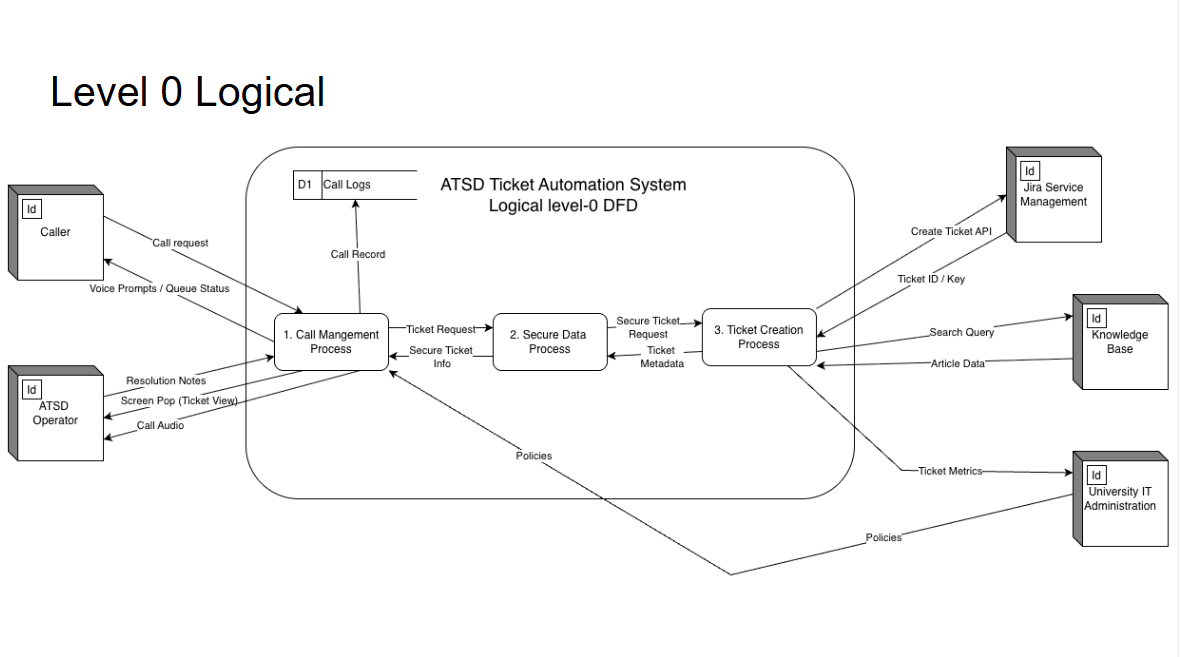
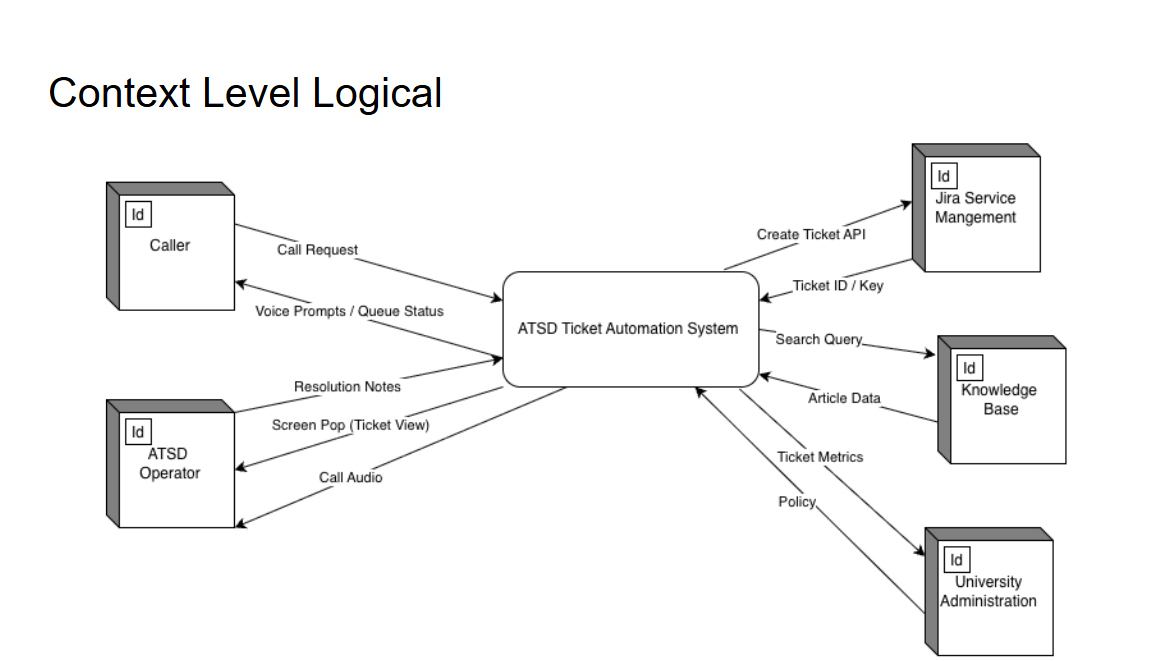
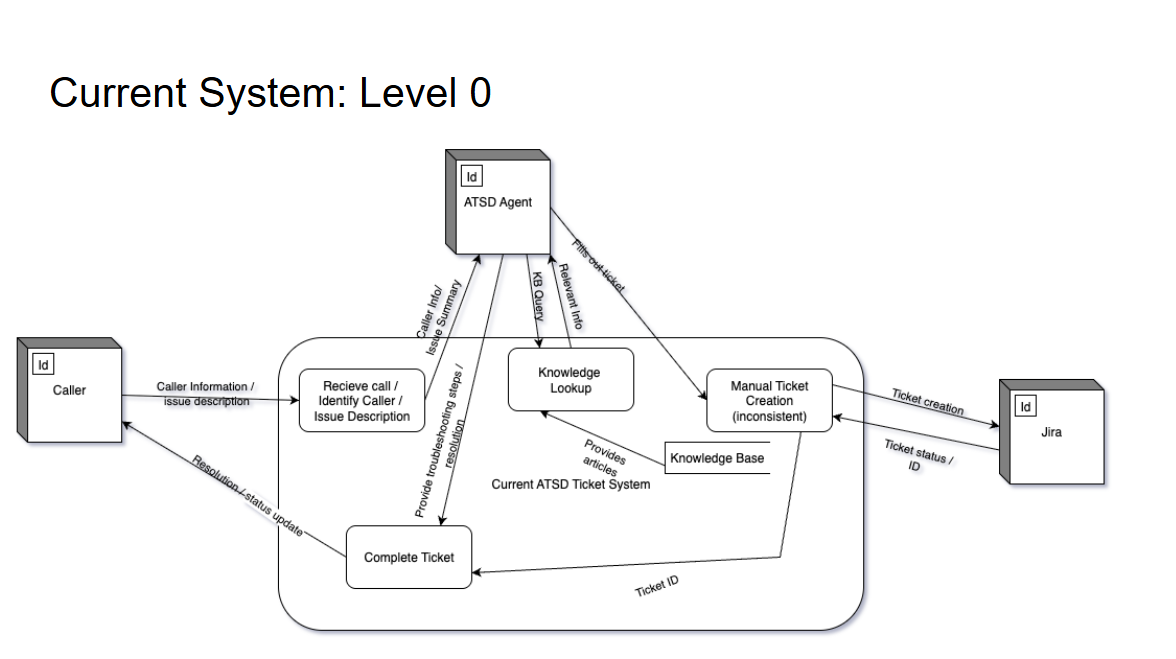
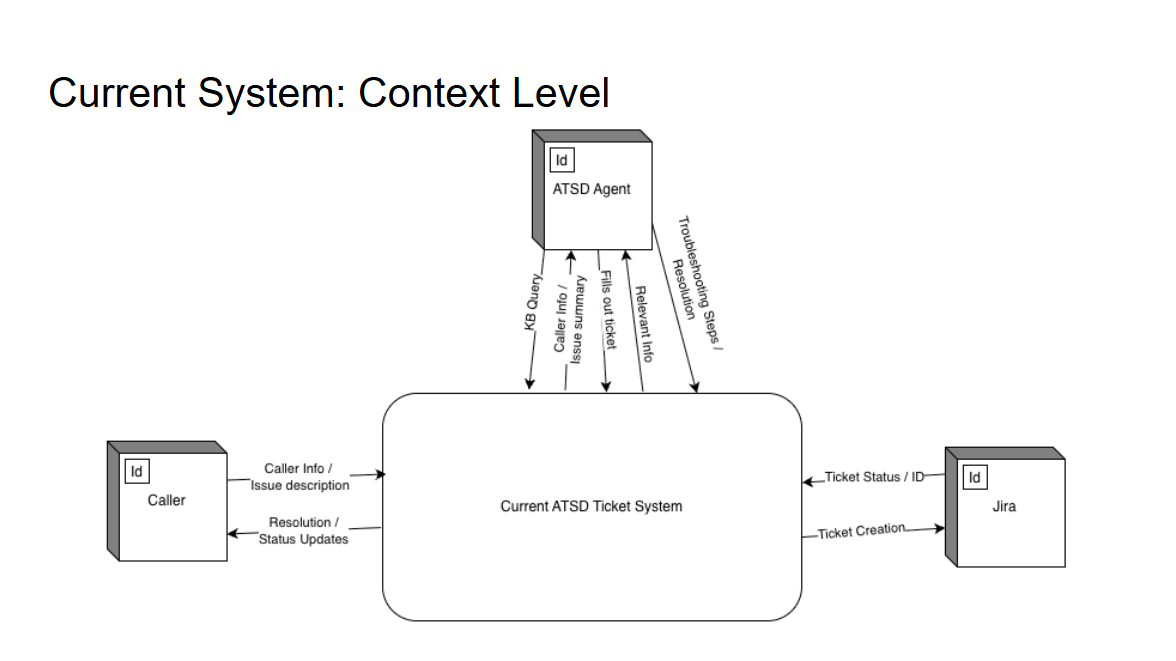
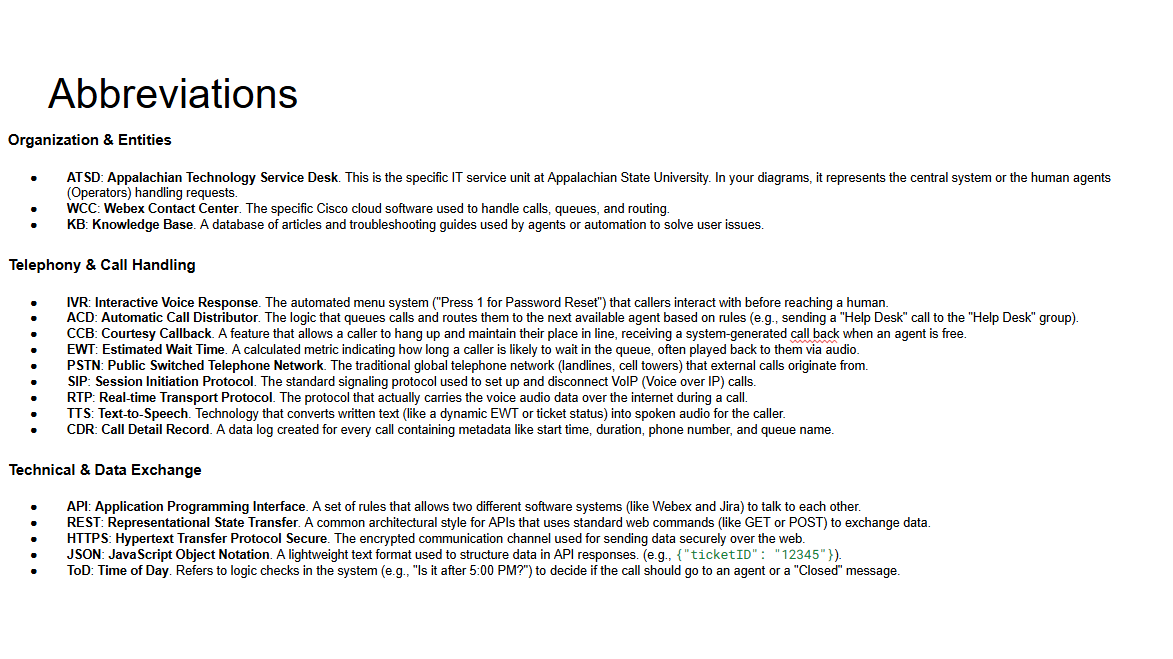
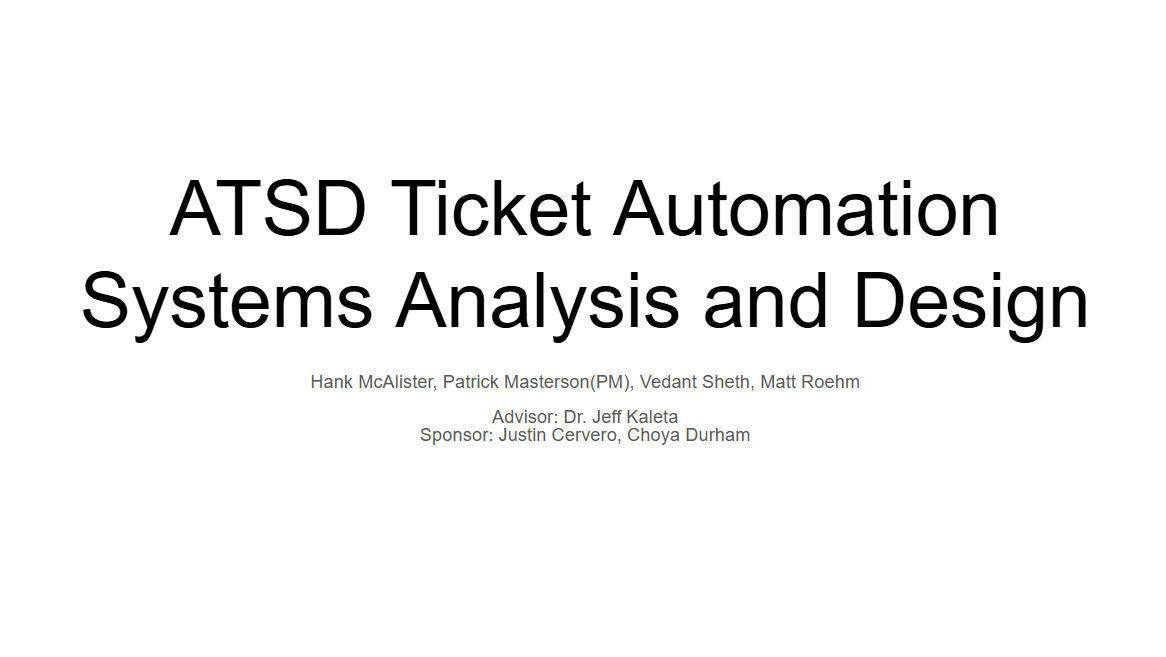
**DELIVERABLE 00 – PROJECT PROPOSAL:**

****

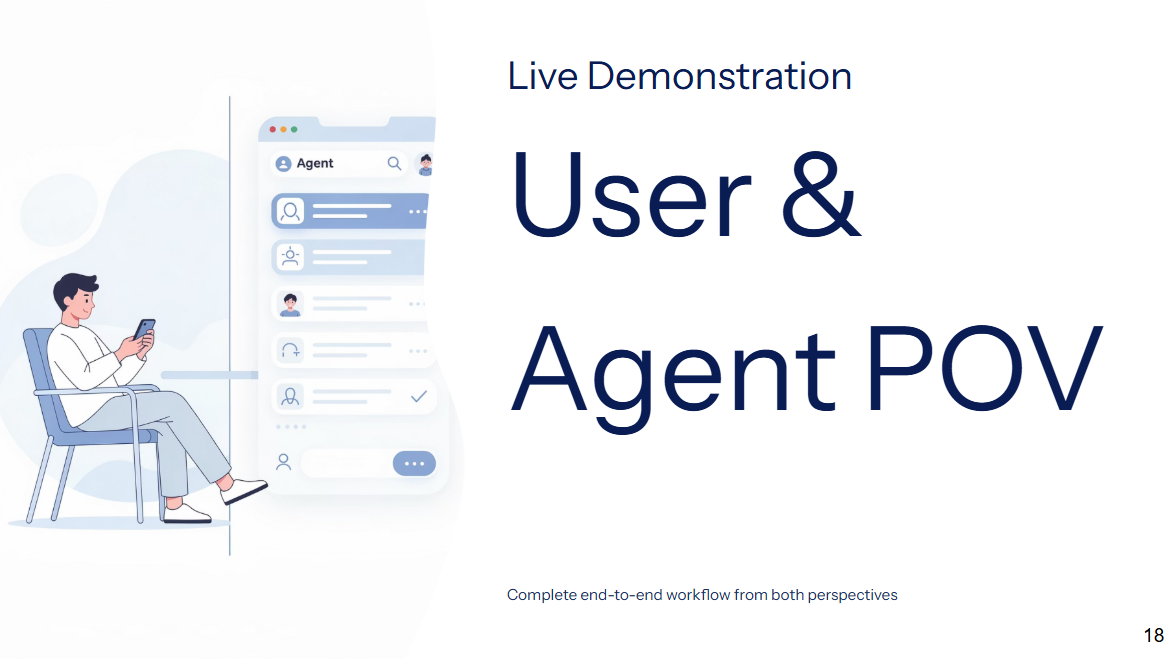
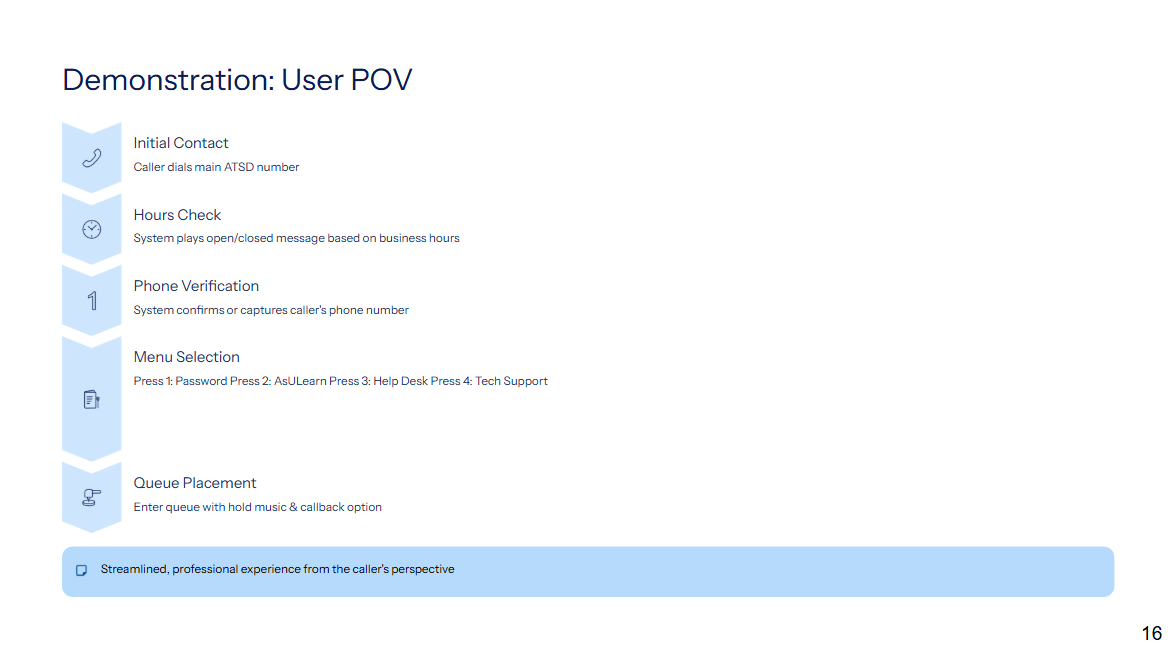
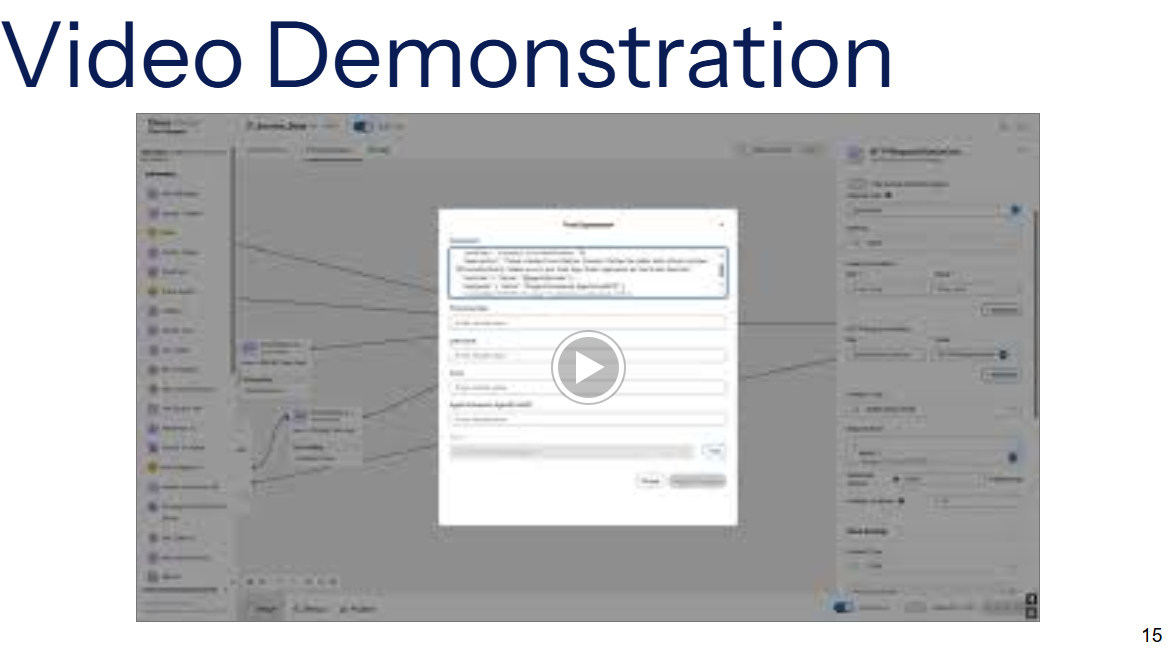
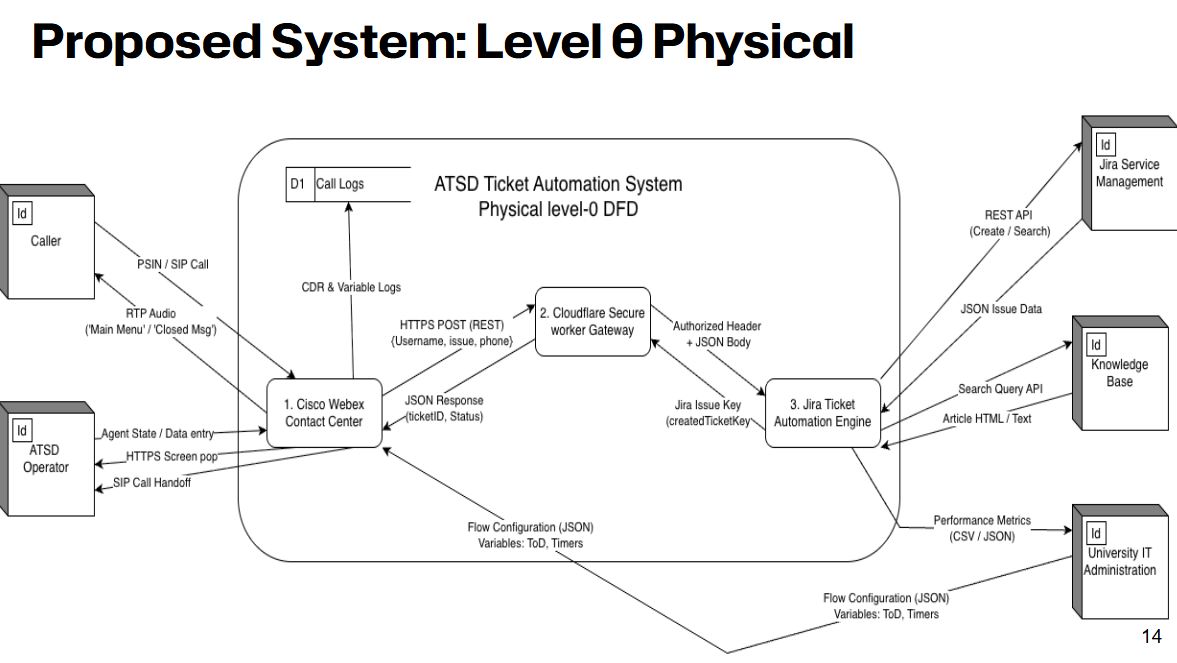
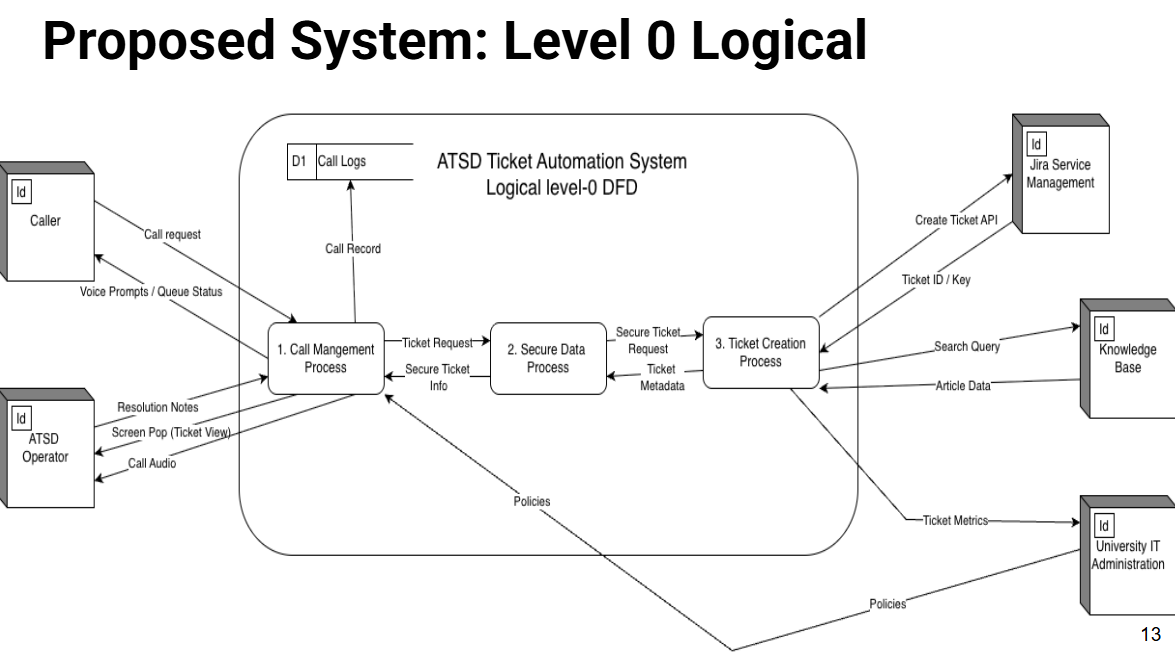
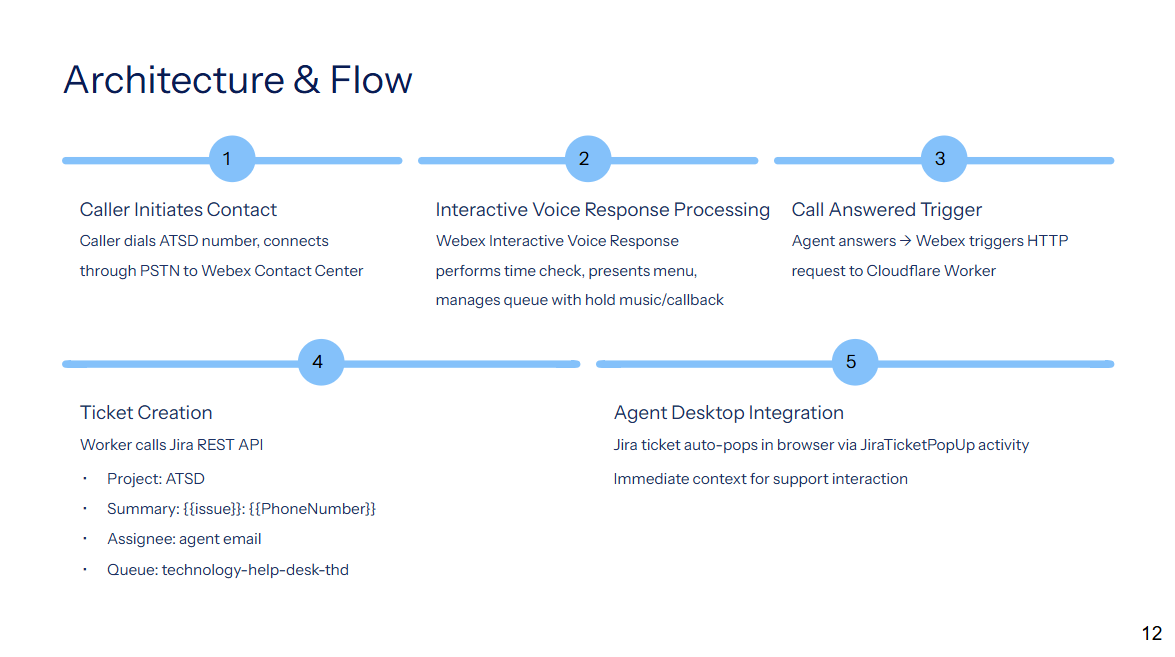
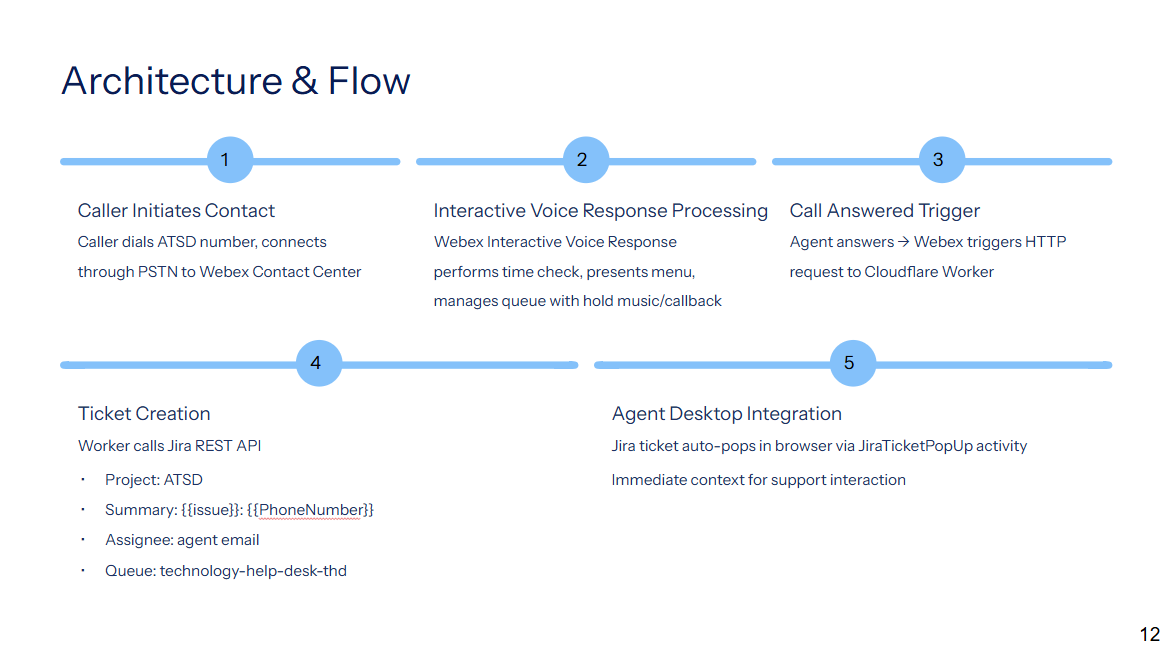
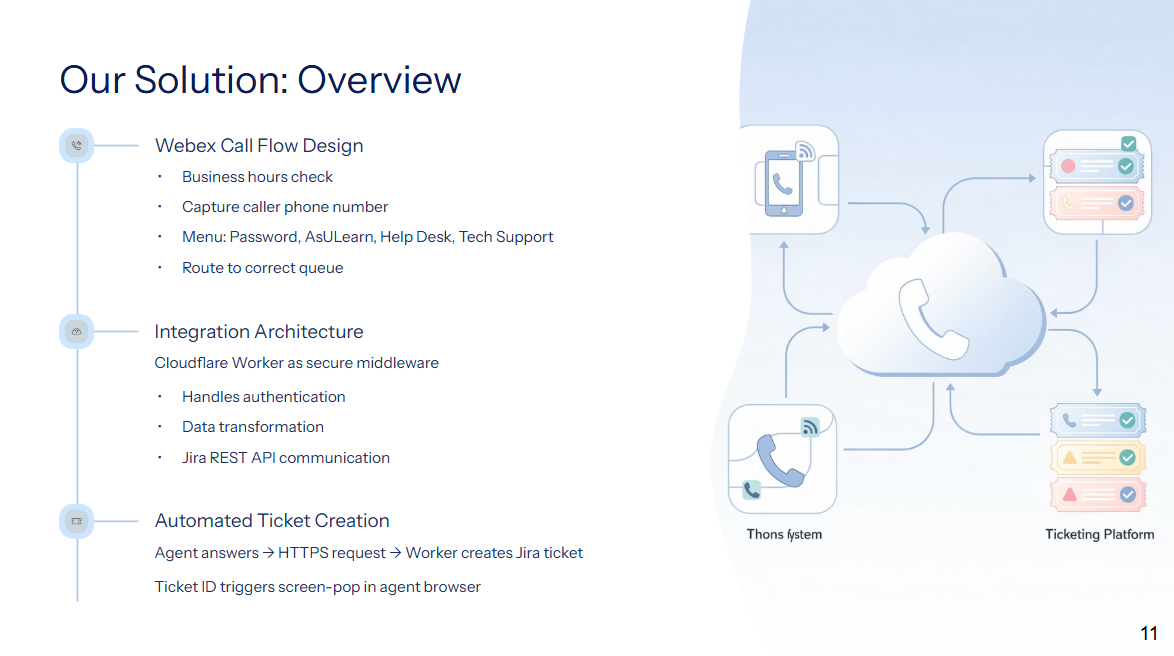
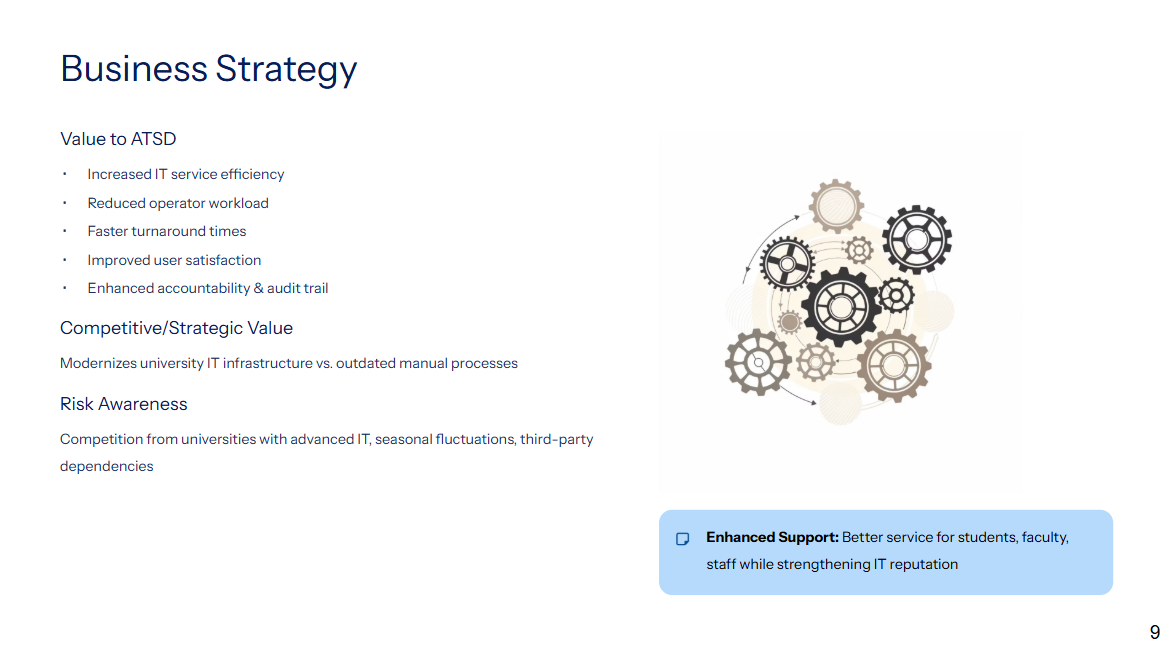
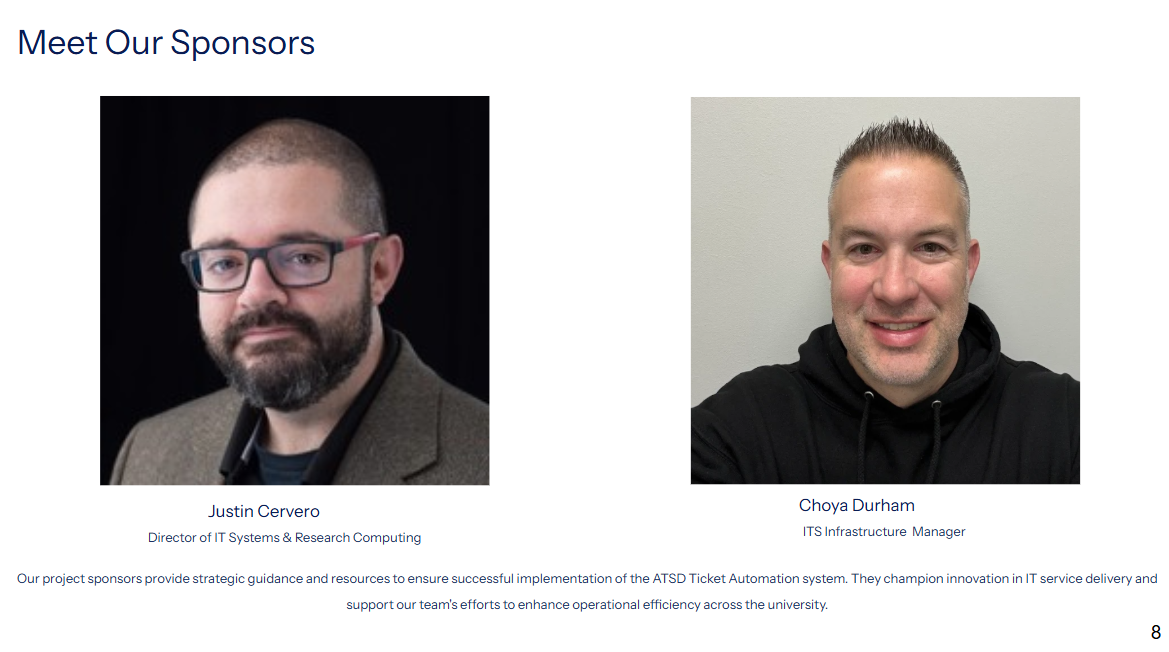
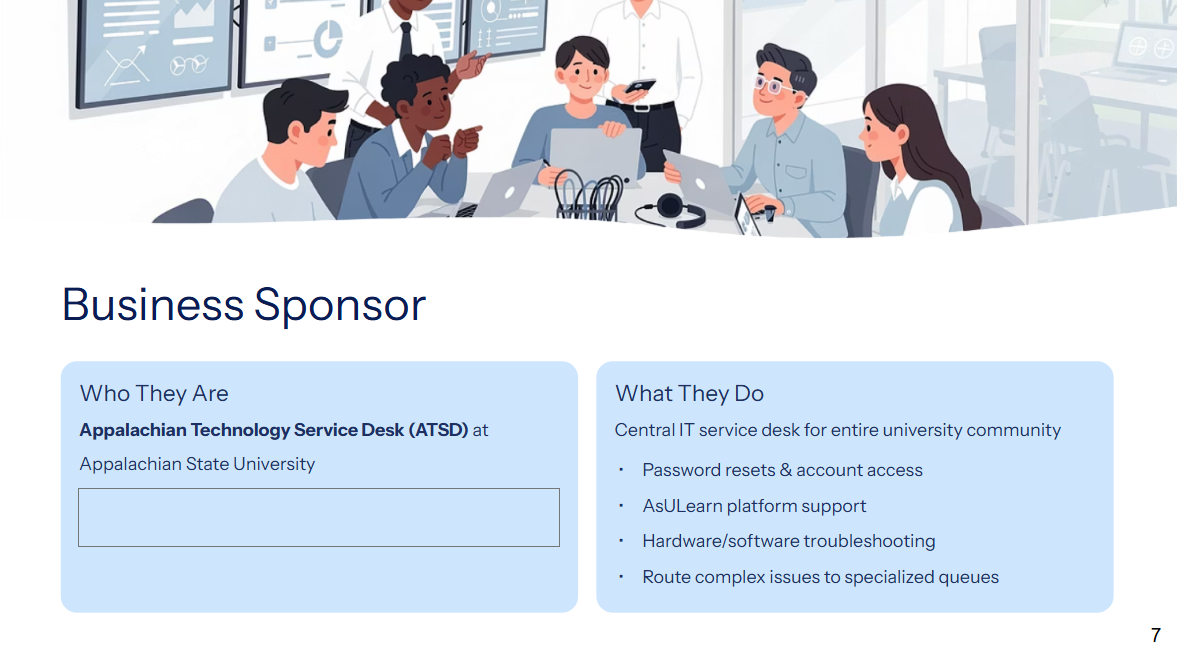
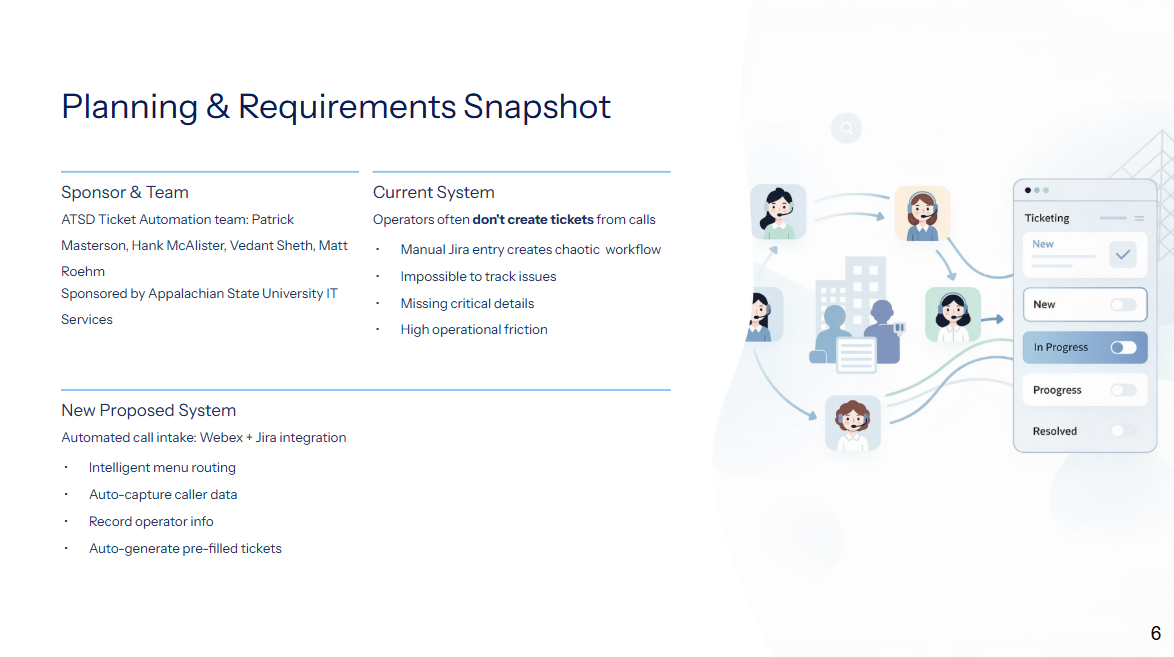
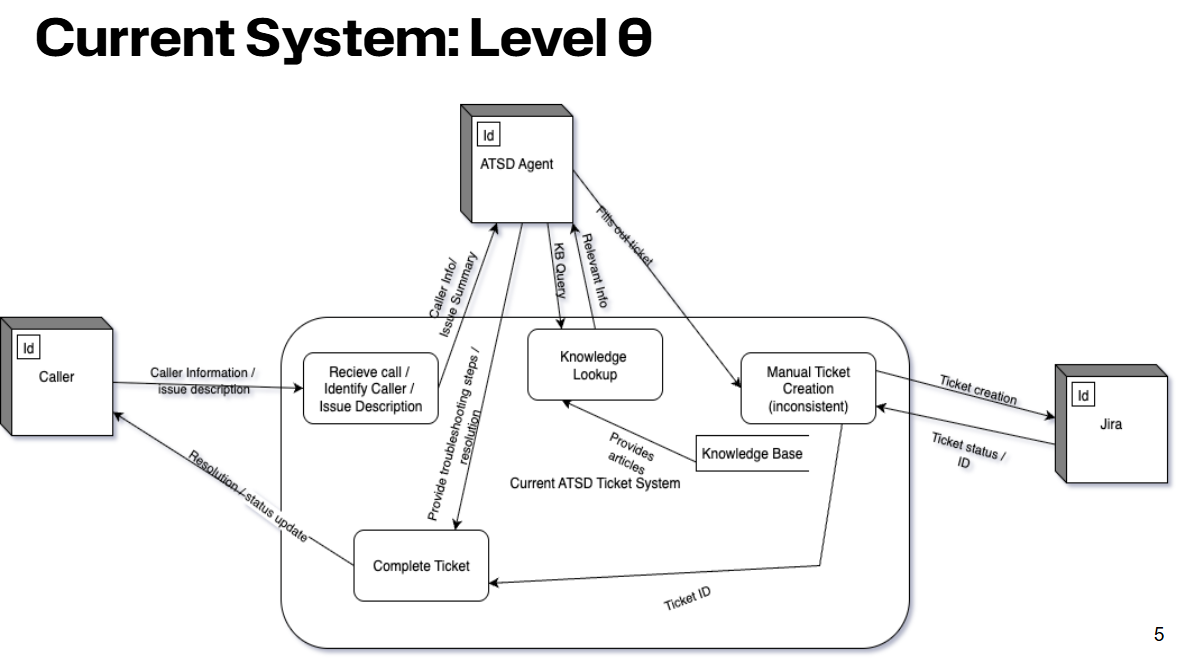
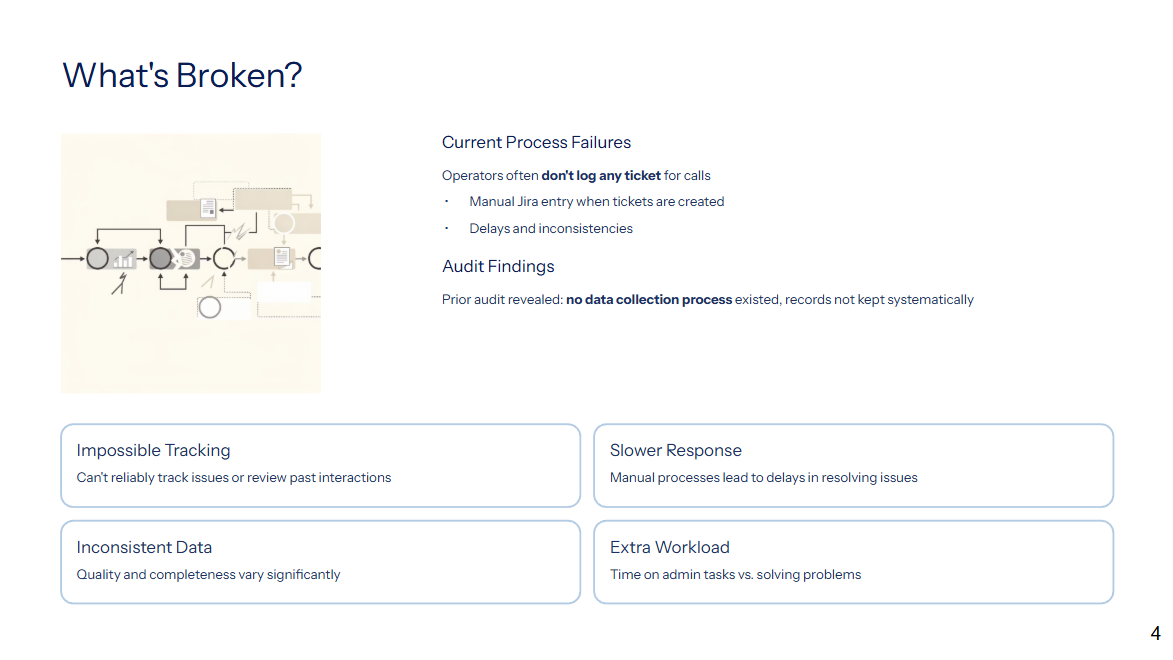
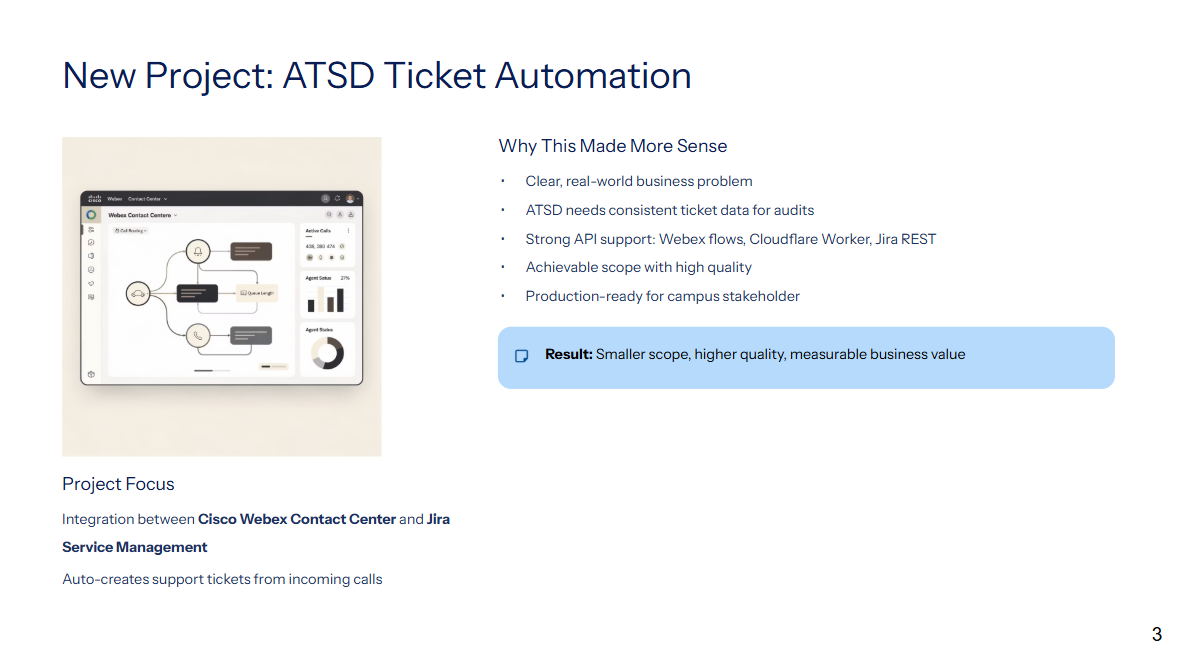
****

****

**DELIVERABLE 02 – ANALYSIS, LOGICAL, AND PHYSICAL DESIGN:**

****

**DELIVERABLE 03 – PHYSICAL SYSTEM DELIVERY:**

****