



**Key Data Flows**

***Chat Request Flow***

1. User submits message through client application
2. Request is authenticated via Azure AD and goes through API Management
3. App Service processes request and establishes SignalR connection
4. App Service retrieves agent configuration from Cosmos DB
5. Request is forwarded to Azure OpenAI Service with streaming enabled
6. Tokens are streamed back through SignalR to client in real-time
7. If tool calls are needed, Azure Functions execute the appropriate tools
8. Authenticated actions use the user's delegated token to make API calls
9. Metrics are logged to Application Insights throughout the process

***Agent Configuration & Deployment Flow***

1. Developer creates/updates agent configuration in development environment
2. Changes are committed to Azure DevOps repository
3. CI/CD pipeline runs offline evaluations against test cases
4. Results are stored and an approval request is created
5. Approvers review changes and evaluation results
6. Upon approval, configuration is stored in Cosmos DB with new version
7. Feature flag is created for gradual rollout (A/B testing)
8. Online evaluations monitor performance in production
9. Based on metrics, rollout percentage is gradually increased

**Key Technical Components**

***Streaming Implementation***

Azure SignalR Service provides WebSocket connections for token-by-token streaming, while Azure OpenAI Service supports streaming API responses.

***Configuration System***

Azure Cosmos DB stores versioned agent configurations including base model, prompts, tool configurations, and parameters.

***Authentication Actions***

Azure AD provides delegated tokens allowing the agent to perform authenticated actions on behalf of users via OAuth 2.0 flows.

***Evaluation Framework***

Combines offline evaluations (pre-deployment) and online metrics (post-deployment) with Azure DevOps for version control and bisecting issues.