



MANAGING DELEGATED ACCESS FOR AI AGENTS

INTERNSHIP

Contact:

Jeroen Hufkens

Company Information:

academy@identit.eu

IdentIT

Veldkant 35a

2550 Kontich

www.identit.eu

IDENTITY SECURITY, GUIDED BY EXPERTISE

IdentIT is an Identity Security Partner that helps organisations regain control over their digital identities. We combine independent advice with deep technical expertise in IAM, IGA and IVIP, and provide continuous guidance to help organisations build a sustainable identity landscape.

Our team currently consists of 40 IdentIT'ers who collectively manage 40 million digital identities. This makes us one of the largest dedicated Identity teams in the Benelux.

IdentIT is part of De Cronos Groep. This gives us access to the expertise, network and opportunities of a large technology group, while allowing us to retain our own focus, culture and way of working.

At IdentIT, you are given the freedom to take initiative, assume responsibility and shape your own growth. At the same time, you are never on your own: we bring together different strengths, help each other move forward and share our knowledge. New ideas and critical questions are always welcome, because learning works both ways. We approach complex challenges with focus and ambition, while valuing openness, humanity and spontaneity just as highly.

Assignment

During this internship, you will build a security assistant for a user's own digital identity: an AI agent that performs tasks on behalf of a signed-in user, with PingAM 8.1 controlling what the agent is allowed to do. AI assistants increasingly perform tasks autonomously for users, making one question essential: who is acting, on whose behalf, and with which permissions? PingAM 8.1 addresses this with a new registration type called AI Agents. Through token exchange, an agent receives its own limited and revocable token, which shows the user on whose behalf it is acting. The focus is on Identity & Access Management, OAuth 2.0 and the Model Context Protocol (MCP), the open standard that enables AI assistants to communicate with other systems.

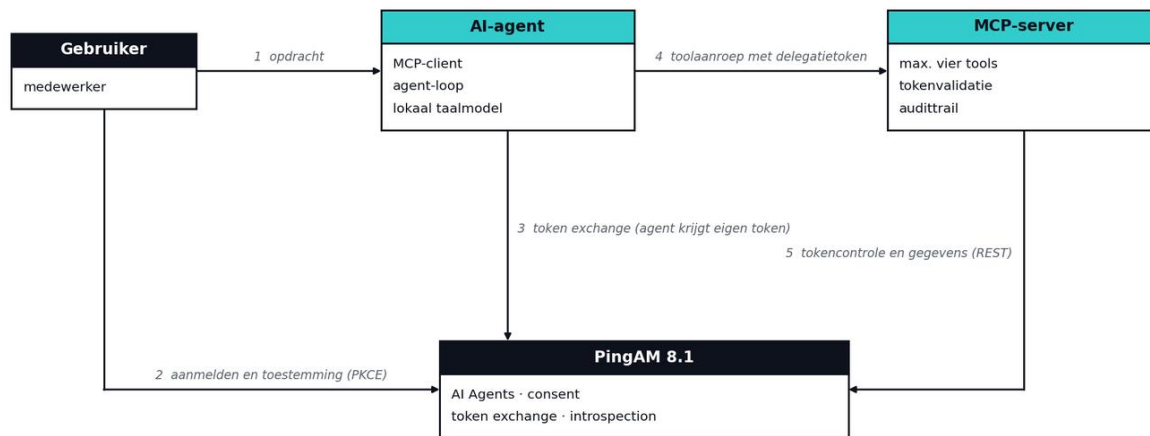
At the heart of the solution is an MCP server: a small web service that offers four functions as tools for an AI assistant. These tools show which apps and agents have access to the account, show where the user is signed in, allow the user to be signed out everywhere, and revoke access. They operate on data already managed by PingAM, so no separate database is required. The language model behind the agent runs locally and free of charge through Ollama. The agent itself can remain simple: the assignment will be assessed primarily on the identity layer.

What will you do?

- Install PingAM 8.1 with PingDS and configure a realm, an OAuth 2.0 provider, a client for the user, and an AI Agent with the associated may act script.
- Build an MCP server with no more than four tools that operate on data managed by PingAM itself: active sessions and granted consents. Additional tools would only broaden the scope of the assignment.
- Have the agent obtain its token through token exchange, so that the act claim shows that it is acting on behalf of the user.
- Have the MCP server validate every tool call with PingAM: is the token authentic, is it still valid, and does it contain the correct permissions (scopes)?
- Require separate, explicit user confirmation for high-impact actions, such as signing out everywhere or revoking access.
- Make the agent's consent revocable with immediate effect.
- Build an audit trail that records for every call which agent, user, tool and permissions were involved, as well as the outcome, including denied requests.
- Build the agent around a local open language model through Ollama, with a scripted fallback mode, and demonstrate the tool calls directly through MCP Inspector, the official testing tool for MCP servers.
- Document the complete solution: prerequisites, configuration, scenarios, troubleshooting and day-to-day operation.

Architecture guidelines

The diagram below shows the basic structure we expect to see in the solution. The numbers trace a single tool call from instruction to audit. The dark elements already exist; you will build the light elements yourself. You will determine the complete implementation during your analysis.



Reference architecture for secure AI agent access.

- Start with the AI Agents registration type in PingAM 8.1 and the associated may act script. This allows you to define in advance which agent may act on behalf of which user.
- PingAM 8.1 with PingDS is the only existing component. PingIDM is not used in this assignment.
- The MCP server validates tokens but does not issue tokens itself or manage passwords.
- Explain and justify any deviations from this structure in your analysis.

What technical skills will you learn?

- OAuth 2.0 and 2.1 in practice: Authorization Code with PKCE, token exchange (RFC 8693), introspection, scopes and least privilege.
- Configuring PingAM 8.1: realms, the OAuth 2.0 provider, clients, the AI Agents registration type and the may act script.
- The Model Context Protocol and its authorisation specification: building an MCP server as a protected resource server, with discovery through protected resource metadata.
- Working locally with open language models through Ollama, including tool calling.
- Understanding prompt injection as an authorisation problem and enforcing security server-side.
- Writing technical documentation, including a comparison between PingAM 8.1's capabilities and the requirements of the MCP authorisation specification.

What soft skills will you develop?

- Communication skills: reporting and documenting clearly and professionally, and collaborating with colleagues and stakeholders.
- Collaboration: working effectively within a multidisciplinary team in an agile environment.
- Problem-solving: analysing technical challenges and resolving them independently or in consultation with others.
- Self-organisation: taking responsibility for your tasks, planning and progress.
- Critical thinking: substantiating your choices and continuously improving based on feedback and evaluation.
- Customer focus: considering user needs and the business context when designing solutions.

Methodology

- The internship assignments follow the Scrum project methodology to provide a structured and efficient approach to tasks and subtasks.
- This reflects IdentiT's approach to projects.
- Scrum is a flexible framework that:
 - Promotes collaboration
 - Encourages adaptability
 - Ensures transparency.
- The work is organised into time-boxed iterations called sprints:
 - Each sprint lasts 2 to 3 weeks.
 - Each sprint has clear objectives and tasks.
- At the start of each sprint:
 - The intern(s) and supervisor(s) hold a sprint planning session.
 - The objectives are defined.
 - A sprint backlog is created.
- During the sprint:
 - Weekly stand-up meetings are held.
 - Progress is discussed.
 - Any obstacles are identified and addressed.
- At the end of each sprint:
 - A sprint review is held.
 - Where possible, a demo is given.
 - Feedback is collected.
- Scrum contributes to a collaborative and adaptable working environment.
- The aim is to achieve the best possible results within the given timeframe.

MVP

The minimum feature set we expect is:

- PingAM 8.1 with a realm, OAuth 2.0 provider, user client and registered AI Agent
- An MCP server with up to four session and consent tools, secured through introspection and scopes
- A local-model agent that obtains a delegation token via token exchange, with a scripted fallback
- Explicit confirmation for high-impact actions and consent that can be revoked immediately
- An audit trail and live demo of the complete flow, including directly in MCP Inspector
- Documentation covering prerequisites, configuration, scenarios and troubleshooting

Objectives

The following deliverables are expected at the end of the assignment:

- Technical analysis of the problem, including a concise comparison of PingAM 8.1 and the MCP authorisation specification.
- The architecture of the required components.
- The MVP and any additional features, with the PingAM configuration as code.
- A demo of the end-to-end solution.
- Documentation of the entire process.