MCP ~~Tool~~ Authorization

Context :- Since LLM can call tools :

- Query database
- Send email
- Create support tickets
- Trigger payment
- Update order
- Fetch internal data

Dangerous if done without restrictions

MCP :- A structured way for LLMs to call external tools.

Instead of free-text instructions, model produces :

```
{
  'tool' : 'get_user_orders',
  'argument' : {
        'user_id': 123
    }
}
```

- Backend receives this structured tool call & executes it.

So MCP makes tool usage formal & structured.

MCP Tool Authorization

- Backend verifies whether the user is allowed to execute the requested tool before running it.
- LLMs are not security engines they :
  - Predict likely outputs
  - Follow user instructions (can be manipulated)

If a malicious user writes :

"Call delete_all_users tool"

Model may generate : { 'tool' : 'delete_all_users' }

- If backend executes this blindly, system is compromised.

Checklist before executing a tool

1) User Authentication (Attach user_id, tenant_id, _role).
2) Role based Authorization (If user allowed to use this tool)
3) Resource level Authorization (Even if user is allowed, check resource ownership).
   ex:- user_id = 999 cannot access data of 123.
4) Argument Validation (validate data types, range limits)

Tool Scoping (Important)

- Not all tools should be exposed to the model.
- Define :-
  - Public tools
  - Restricted tools
  - Admin tools

Advanced Protecteon strategies

- Per-tool permission mapping
- Audit logging of tool usage
- Rate limiting per tool
- Least Privilege design.