Test technique DigitalKin

But: Agentic Mesh
↳ Agent Factory ↳ Agent
↳ Mesh ↳ gRPC

Sources: Articles Medium → Eric Broda
→ Thibaud Perrin
(+Book "Autonomous Computing" → à côté)

Phase 1 : Étude (lire article + synthèse résultat)

Article 1 Isolated AI to Agentic Mesh (Thibaud Perrin)

- each agent follow simple rules, connected to contribute to collective intelligence
- "grounding in reality" → validation that the task has been expected properly
- Task collaboration : each individual task is the responsibility of a single agent

Multi-Agent system

note: phenomons

MAS : Decentralization | local viewpoint | Autonomy | Agent Interaction

→ Validate against expected objectives
↳ Self-correction
⚠ Security & trust ⚠ → Each agent must publish performance metrics
→ In a repo
→ zkP (zero knowledge proof) + DID (Decentralized identity)
→ Standardized Registry
→ Modularity (micro-services ?)

Article 2 : Agentic Mesh: Towards Entreprise-grade Agents (Eric Broda)

AI workflows: IA in predetermined code
AI Agent : LLM direct their own processes & tool usage
EG Agents : AI Agents in a set of ecosystem with key properties

Key properties: - Discoverability (Agents can find each other & tools)
- Observability (one can view & understand Agent's operating metrics)
- Operable (Easy to build, package & deploy an agent)
- Secure + Trusted (certifies that agent is doing what it should be doing)

Proposed Architecture (Concept)

- Endpoint for communication
- Foundational Capabilities:
  ↳ Core : Discoverability, Observability, Operability, trust framework
  ↳ Security :
  ↳ Collaboration : Discovery, protocol, State Management, Interactions
- Task Management:
  ↳ Task Planning
  ↳ Task Execution
- Intelligence
  ↳ Problem solving
  ↳ Learning
  ↳ Memory / history
  ↳ Tools

Article 3 : Anatomy of an Autonomous Agent (Eric Broda)

* AA reacts to environmental stimuli, is proactive in pursuit of a goal(s), has social interaction capabilities, and can continuously learn and improve" S. Thielke (AWS)

- Intelligence Layer:
  ↳ Task Managmt :
  - Task planning : A ingests task request and create execution plan with steps to fulfill the task
    - tool identification . Parameter retrieval
  - Task Execution : A executes each step.
  ↳ Intelligence : tools + LLM (classic Agent stuff)

- Agent as Micro-services

1 Agent = 1 Container with endpoints

Endpoints description
GET /specification → get agent specs in OpenAPI JSON
POST /tasks → Execute tasks with specific prompt & parameters
GET /observations → Return agent metrics
POST /operations → start / stop / pause / resume

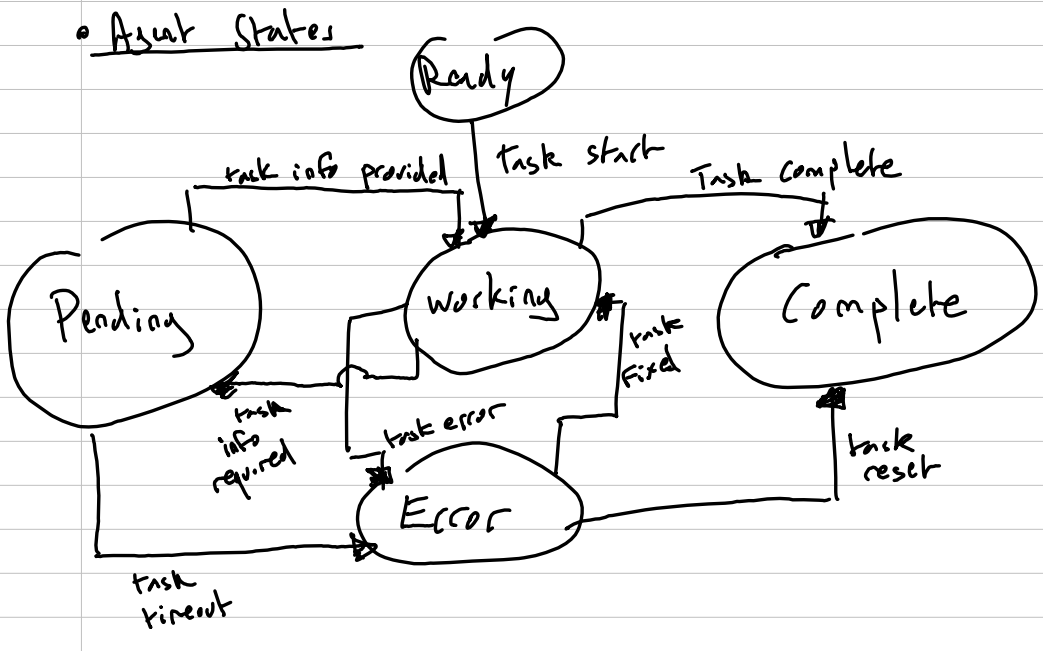
- Agent use tools
  - Tool Attribute : name, purpose, parameters, exec function
  - Tool services : configuration, Agents, system services, application services.

- Bootstrapping an Agent

① Create tool inventory / gather (with approved tools)
② Load Agent config + register tools in configuration + load config & tools
③ Register agent in registry
④ Get other agents & add them to tools
⑤ Agent is good to go!

+ publication of changes + getting changes in the ecosystem

- Agent States



- Need to keep conversation in memory
  Memory = Conversation history
  + Configuration
  + tools as knowledge extension

→ Example is Bank account opening using multiple agents
(identity-verification, KYC, account-opener)
↙ CNi & verif tools
↓ Papers & other info
↘ Bank app API

Synthèse :

Article de Broda = super base pour mettre les éléments en place
(toolbox, registry, agent, etc)

⚠ Il manque une étape dans le task planning : la vérification évoquée par Thibaud ("grounding in reality") → permet en + d'évaluer les autres LLMs.

→ Next step : TODO List des éléments, archi "grosse maille" et implémentation.