AGENTIC BY DESIGN

CONVERTEO

Télécom IA · 04.06.2026

Converteo, cabinet de conseil hybride et *pionnier français de l'IA agentique*

Nous sommes alliés aux principaux players IA et Tech mondiaux pour assurer le succès de nos clients.

Partenariat GCP First

Avec Google Cloud Platform, nous bénéficions d'un accès privilégié aux dernières innovations Google. Du POC au déploiement, nous construisons nos pipelines de production dans un écosystème performant et scalable.

Un Lab IA pour l'innovation

Nous catalysons notre propre transformation avant de la porter chez nos clients. Le Lab IA Converteo mène des projets réplicables via des POC agiles et structurés, adhérant à notre charte WAA : Well-Architected Agentic.

Expertise de pointe

Converteo compte 450 collaborateur-ice-s tech et business, dont un Hub Tech de **200+ ingénieur-e-s expert-e-s et certifié-e-s en data, IA et agentique.**

Google Cloud · Gemini · Microsoft Azure · OpenAI · aws · DUST · ANTHROP\C · Google Marketing Platform · Amplitude · CONTENT SQUARE · snowflake

Notre expertise pour la transformation agentique

AI ENGINEERING

Concevoir et déployer des systèmes agentiques de bout en bout pour la productivité et l'expérience utilisateur

- **Model Selection & Config.** — Select, fine-tune, prompt and manage lifecycle
- **Agent Design** — Orchestrate, route, manage memory and state
- **Reliability & Agent Governance** — Ensure security, compliance & explainability
- **Data Preparation** — Transform raw data into context
- **Knowledge base** — RAG, structuring, and grounding
- **Production Supervision** — CI/CD for models, drift detection, and observability

PLATFORM ENGINEERING

Construire les piliers data Cloud de produits agentiques et tech *AI-ready from day one*

- **Data Platform & Cloud Infra** — APIs, semantic layer and vector stores
- **Data apps. & AI integration** — APIs, interfaces, and integration of AI models into products
- **Business modeling & governance** — Semantic layer, data governance
- **Analysis & Conversational BI** — Restitution, Natural Language, and Agentic Analysis
- **Data Quality & Traceability** — Pipelines, anomaly detection, traceability
- **Production Supervision** — CI/CD for models, drift detection, and observability

MEASUREMENT ENGINEERING

Capturer une donnée comportementale fiable à l'ère agentique via des sources de vérité uniques

- **AI readiness** — LLM signals, agentic instrumentation, input data
- **Server-side architecture** — Event collection, server tagging & routing
- **Product Analytics** — Instrumentation, behavioral analysis & experimentation
- **Web Performance** — Core Web Vitals & Technical Optimization
- **Privacy & Compliance** — Regulatory compliance, consent & governance
- **Data Activation & First-Party Signal** — Connection to CDPs/DSPs/CRMs and clean rooms

Travail d'équipe !

Nos expertises sont poreuses, elles se complètent.
Chez Converteo, pas de succès sans coopération.

Vos intervenant-e-s

Léane Salais, AI Engineer

- Master IA Sorbonne Université (MIND)
- Ex-chercheuse, profil interdisciplinaire
- Formatrice en data et IA
- **Mon credo : quelle que soit votre expertise, jamais de projet sans étude du contexte !**

Mes clients chez Converteo

CRITEO · weavenn · KERING · adeo

Brice Van Aken, Data Engineer

- Ingénieur de l'UTBM
- Issu système réseau, sécurité
- Forte appétence Cloud, Data, IA
- **MacGyver dans l'âme : peu importe le domaine, tout n'est qu'une déclinaison du système. Je m'adapte.**

Mes clients chez Converteo

FNAC DARTY · ACCOR

En duo chez Converteo, nous animons le Genius Bar, rendez-vous de vulgarisation et de **montée en compétences collective.**

Au programme...

...une montée en compétences adaptée à vos besoins, qui vous donnera les clés pour devenir des professionnel-le-s de l'agentique

Explorer

Un premier contact aujourd'hui

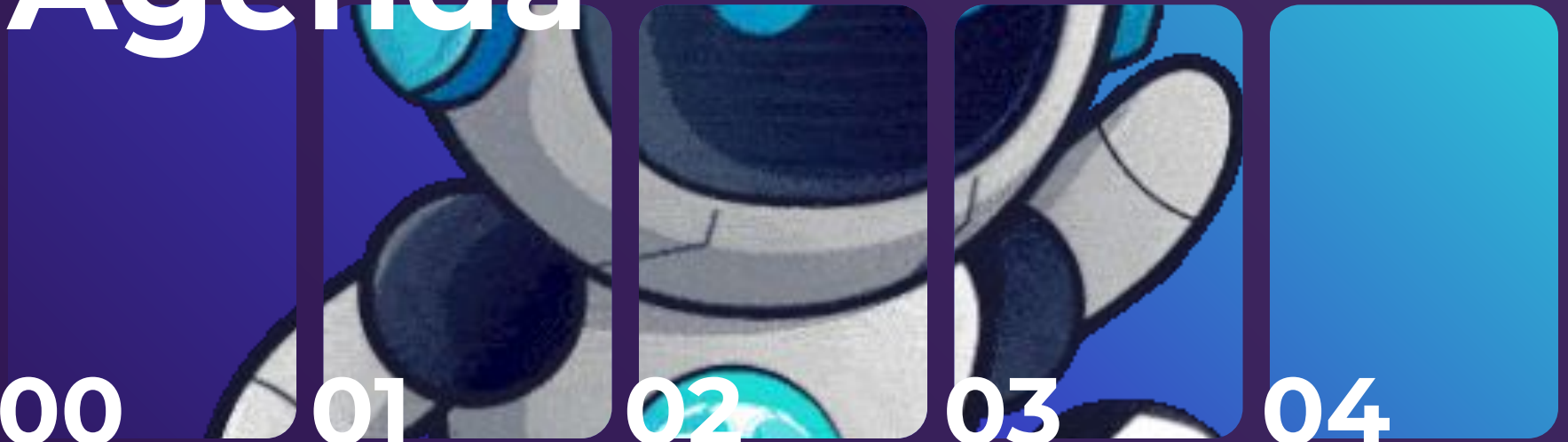
Cadrer le sujet, connaître les outils, comprendre notre méthode et répondre à vos questions sur la théorie.

Appliquer

Vers l'autonomie : un cas pratique à l'automne

Pouvoir se faire la main sur les outils GCP, pas à pas, avec des cas clients réels et une exécution guidée.

Agenda



00

L’IA… agentique ?

Les bons repères pour parler d’agents

01

Faire de l’agentique en 2026

Les changements récents du paysage

02

C’est quoi, être GCP-first ?

Ce qu’apporte notre partenariat avec Google

03

Et concrètement, comment ça tourne ?

Penser pratique en attendant septembre

04

Petit quizz

(parce qu’on est quand même en école)

00

L’IA… *agentique ?*

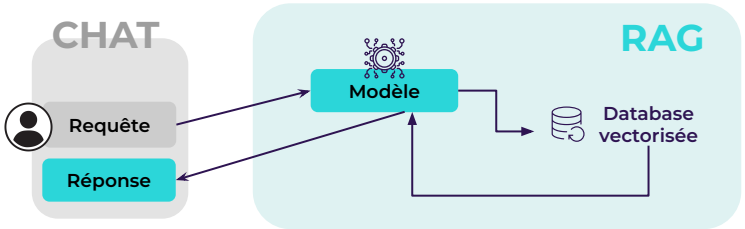
L'IA générative était douée, mais encore limitée...

Le RAG (Retrieval Augmented Generation) comme première amélioration des LLMs

Les premiers systèmes de LLM étaient isolés : ils avaient des connaissances figées dans le temps via leurs données d'entraînement, mais ignoraient tout des actualités et surtout des données d'entreprise.

Introduit en 2020, le **RAG** agit comme un pont entre le modèle raisonnant et des sources de données définies.

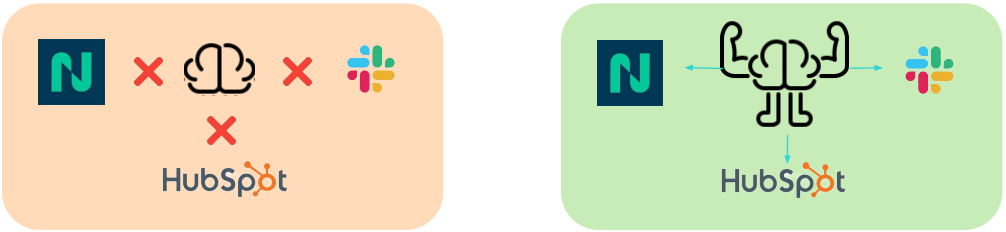
Mais même avec le RAG, l'IA reste incapable d'agir et ainsi de simplifier nos workflows.



L'IA générative était douée, mais encore limitée...

L'agentique commence en faisant *interagir* l'IAG avec des apps extérieures

Pour dépasser cette limite, on donne au modèle la capacité d'utiliser des **outils** (tools) :
il peut donc explorer son environnement pour accomplir sa mission, par exemple en communiquant via des appels API.



API (Application Programming Interface)

Interface de programmation qui permet à des applications de communiquer entre elles pour des fonctions précises.
ex : l'API de Gmail permet d'utiliser `envoyer_un_email` *ou* `lire_les_emails_recus`

**L'agent sélectionne cette API parmi les options en raisonnant sur son besoin,
il génère un appel de fonction avec les paramètres nécessaires,
et exécute cet appel pour réaliser l'action dans le système externe.**

Donc, contrairement à une IA générative vanille qui se contenterait de répondre,

un **agent IA** entreprend des **actions** et résout des **problèmes à plusieurs étapes** de manière autonome.

Interconnexions : outils, sources de données, autres agents

Multimodalité I/O des LLM : texte, multimédia

Raisonnement des modèles LLM couplé au design de l'agent

Gemini 3.1

GPT-5

Claude Design

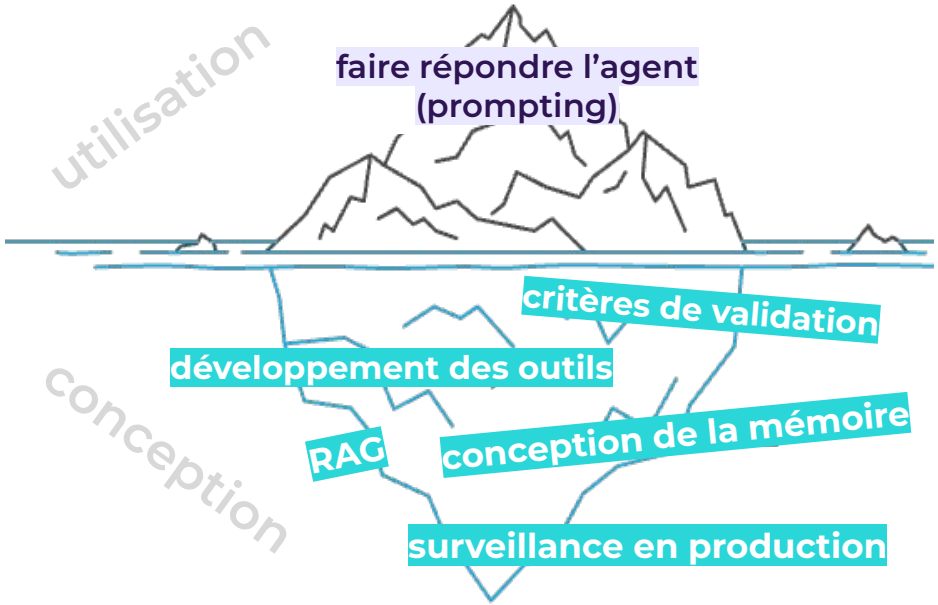
...

Converteo identifie 5 niveaux de maturité agentique

1	**MODÈLE ISOLÉ**	Un chatbot se **contente de répondre aux questions**. **Aucune action** sur des systèmes externes.
2	**AJOUT D'OUTILS (AGENT)**	Le modèle **peut utiliser un ou plusieurs outils** pour des **actions simples** (envoi mail / recherche doc).
3	**AGENT À ÉTAPES**	L'agent **peut décomposer un objectif complexe** en un **plan actionnable** et l'**exécuter en autonomie**.
4	**SYSTÈME MULTI-AGENTS**	Plusieurs **agents spécialisés formant une équipe** pour résoudre un problème. **1 agent = 1 rôle défini.**
5	**BOUCLE D'AMÉLIORATION**	Les agents **apprennent de leurs actions** (feedback) puis **s'auto-corrigent** pour leurs réponses futures.

*Aujourd'hui, la plupart de nos clients se situent entre **2** et **3***

Votre futur rôle



Pour un utilisateur, maîtriser l'art du prompting, c'est savoir poser la bonne question pour garantir la fluidité des interactions, la qualité des réponses et le succès des tâches dévolues à l'agent.

Mais pour porter la révolution agentique jusque chez nos clients, nous devons :

- **spécifier le comportement** de l'agent
- **définir son accès aux connaissances** nécessaires pour son rôle (gestion du contexte, de la mémoire et des outils)
- **évaluer sa performance** en production

Les éléments du design d'agents

Raisonnement
La capacité de l'agent à analyser et à prendre des décisions
→ Quelles règles ?

Action
La capacité de l'agent à interagir avec son environnement
→ Quels outils ?

Perception
La capacité de l'agent à comprendre son environnement
→ Quelles sources ?

Gestion d'état
La capacité de l'agent à maintenir ses souvenirs et sa liste de tâches
→ Quel design de mémoire ?

Objectif
L'objectif clair qui guide les actions de l'agent
→ Quel KPI métier ?

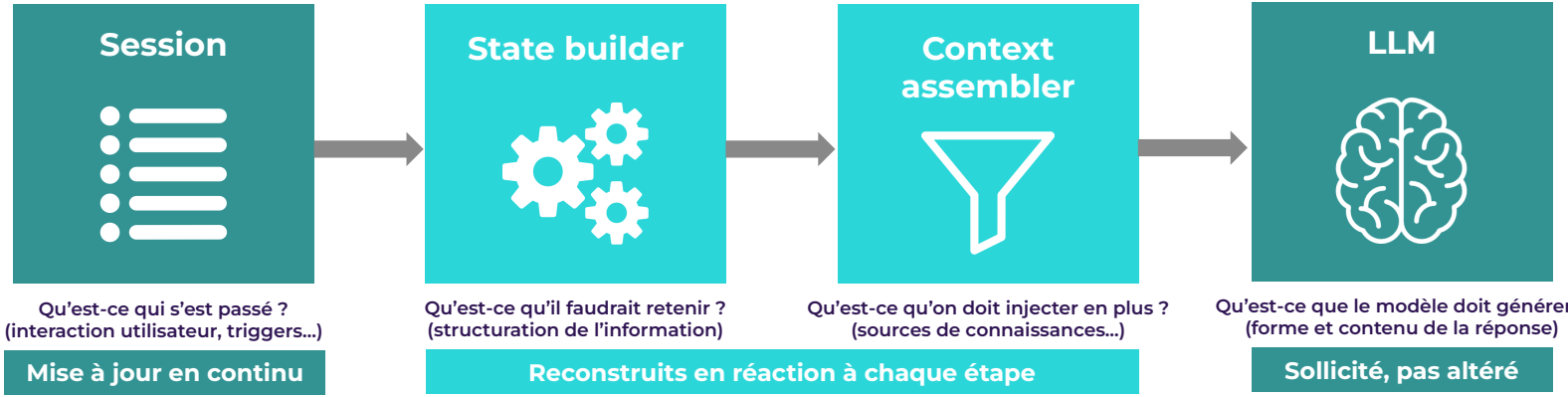
Sécurité
Les limites imposées pour éviter les situations critiques
→ Quels garde-fous ?

Approbations humaines

L'échange avec un LLM au coeur de l'agent

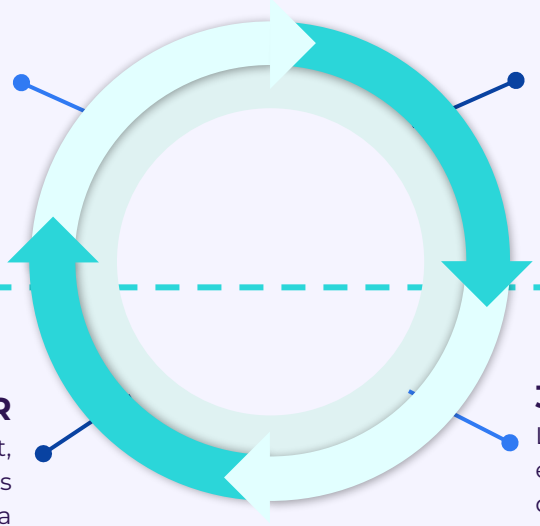
Le contexte : l'art de filtrer l'information pour orienter la génération

On extrait un **contexte** (connaissances structurées)
en empilant des **états** (variables, constructs intermédiaires)
à partir de l'information exhaustive d'une **session** de conversation.
Le concepteur construit la pipeline qui mène de la session au contexte puis envoie cela au LLM.



Le contexte est une sélection dynamique d'information que vous devez penser comme l'apport minimal utile.

La boucle cognitive



1. SOURCER

Une fois le contexte construit au niveau infra, l'agent traite les données hétérogènes qu'il y trouve **sans ingérence humaine**.

2. PLANIFIER/SUGGÉRER

Dans les limites imposées par son playbook ou son design général, l'agent décide d'une stratégie pour atteindre un objectif ou répondre à une question.

IA Générative

3. AGIR

L'agent exécute le plan choisi ; par exemple, un appel API pour récupérer des données ou s'interfacer avec un utilisateur.

4. RÉAJUSTER

L'agent évalue le résultat, y détecte des erreurs et nouvelles idées, et utilise cela pour informer la prochaine itération.

Cette boucle permet à l'agent de percevoir son environnement, de planifier ses actions et de s'auto-corriger pour atteindre son objectif en autonomie.

Une hiérarchie dans le contexte

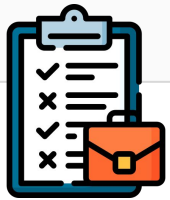
La notion de system prompt

L'agent a à disposition une mémoire court terme (connaissance de l'utilisateur, situation ponctuelle, etc.) mais aussi une mémoire long terme (qui il est, sa mission) qui préside à toutes ses décisions.

System Prompt (Mémoire long terme)

On injecte des informations dans le system prompt, une instruction fondamentale qui ne changera jamais d'une conversation à l'autre : rôle, persona, format attendu...

C'est une autorité forte pour l'agent qui en suit rigoureusement le contenu, et son design (quoi / où) est essentiel dans le projet.



Historique (Mémoire court terme)

L'agent enregistre les prompts et fichiers partagés *dans la session* et s'en sert pour se souvenir de ce qui a été dit ou fait pendant l'interaction.

La flexibilité de la mémoire aide l'agent à s'adapter à chaque situation, mais attention au bruit et aux failles de sécurité (prompt injection, instruction override).

Rebelote : un LLM au coeur de l'agent

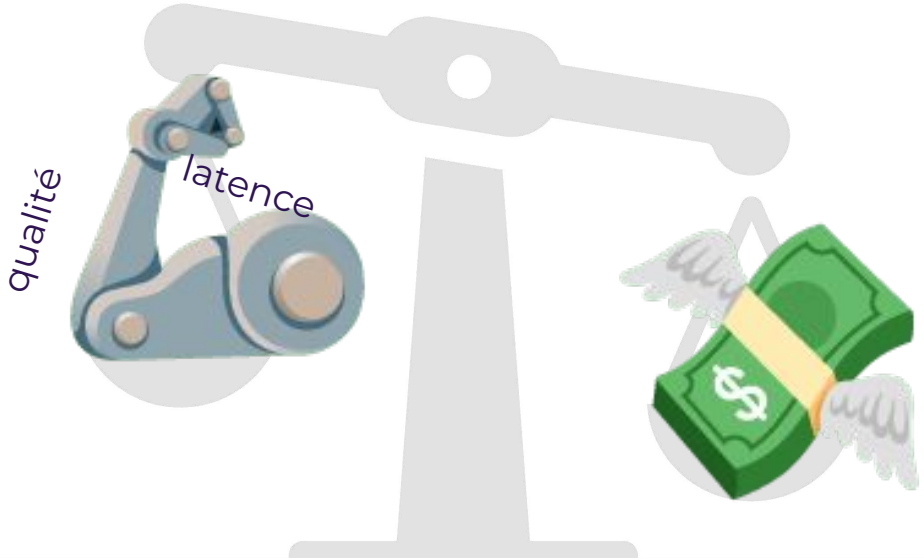
Les fenêtres de contexte imposent un compromis coût / performance. Ne surnourrissez pas votre agent !

Conversationnel
(historique)

Métier
(règles, process, politiques)

Documentaire
(docs internes..)

Transactionnel
(données CRM / ERP..)



Le contexte doit être assez riche, mais pas obnubilant :
sinon, c'est trop de tokens à digérer pour le LLM (💸) et même une garantie de déconcentration pour l'agent.
C'est pourquoi on introduit des sources d'information, mais aussi des mécanismes de sélection et d'oubli volontaire.

Pour vos agents, plusieurs rôles possibles

	Copilote documentaire	Analyste augmenté	Callbot	Enrichisseur de données
	Q&A sur base interne à l'entreprise	Requête de données, insights, recommandation	Répond aux questions et prend des actions	Extrapole des données à partir de sources multiples
Quand l'utiliser ?	Base de connaissances interne et définie	Interaction avec données structurées	Interaction vocale et actions multi-systèmes	Interaction avec données structurées
Quand l'éviter ?	Tâche sur système externe ou analyse complexe	Tâches ouvertes avec plan en plusieurs étapes	Tâches définies par un arbre décisionnel ou au contraire sans playbook	Ingestion classique de données

Les agents peuvent aussi communiquer entre eux

On parle de système multi-agents, et il faut alors une plateforme commune et un système d'orchestration

Chaining

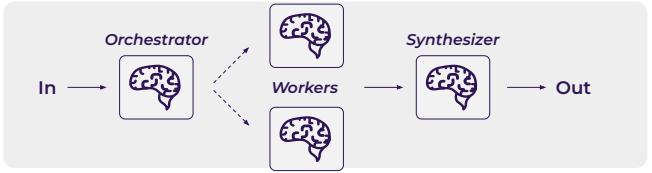
Exécution séquentielle, la sortie d'un agent est l'entrée du suivant

Routing

Un agent routeur dirige les requêtes vers le bon agent spécialisé (à choisir quand il faut un point d'entrée unique)

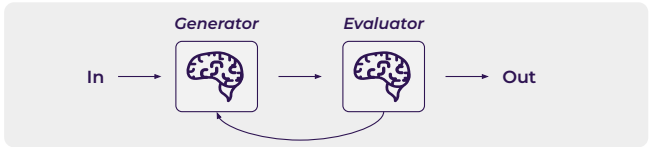
Orchestrateur central

Un agent coordinateur gère le workflow et délègue à des subordonnés (à choisir quand le workflow est complexe et versatile)



Evaluateur

Un ou plusieurs agents valident les sorties d'autres agents



Outiller un agent

Une capacité ajoutée (en général 3 à 10 max par agent) et documentée avec un mode d'emploi

TOOL

Compétence ajoutée à l'agent pour accéder à l'information (base de données, fichier..) et agir sur un système (envoyer mail, créer ticket JIRA). Souvent une encapsulation d'un call API.

Exemple

- Tool d'analyse de doc
- Tool d'extraction de feuille de temps
- Tool de résumé de texte
- Tool pour requêter sur Google Analytics

Une capacité

+

TOOL CALLING

Mécanisme de décision de l'agent quand il cherche à utiliser les tools en vue d'accomplir des tâches. A formater en fonction du LLM cible.

Exemple

Pour trouver la date de livraison du projet Digital Xceleration, l'agent effectue un "tool calling" pour appeler l'outil get_project_details dans JIRA et récupérer l'information

La réflexion pour l'invoquer

Outiller un agent

Une capacité ajoutée (en général 3 à 10 max par agent) et documentée avec un mode d'emploi

Le tool calling est l'affaire de l'agent. Par contre, un tool mal décrit / mal documenté sera un tool mal utilisé !

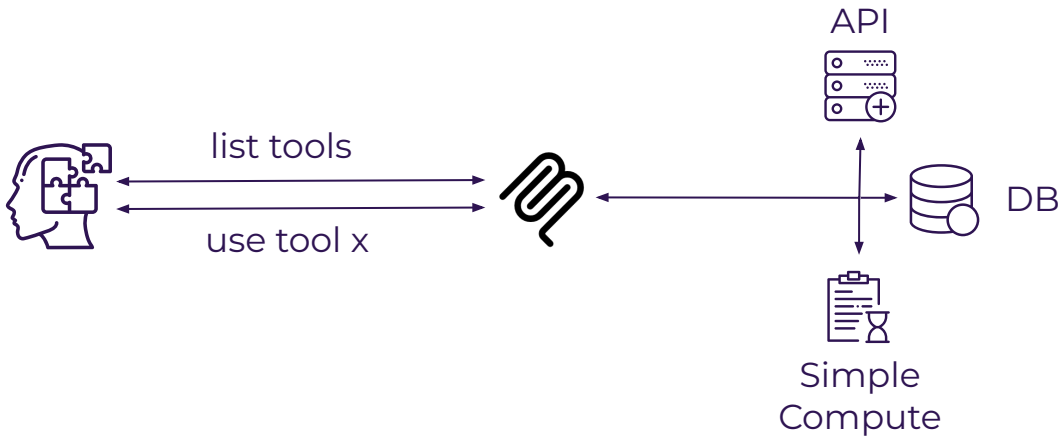
Une capacité

Les serveurs MCP

Comment peut-on maintenir les tools dans la durée ou assurer leur portabilité sur un autre agent ?

Model Context Protocol

MCP est un protocole de communication standardisé qui permet à un agent de découvrir et d'utiliser des outils, sans intégration spécifique ni problèmes de compatibilité inter-LLMs.



Les serveurs MCP

Comprendre l'interaction MCP

Model Context Protocol

1. Les serveurs MCP listent des outils disponibles depuis des sources publiques ou privées.
2. Le concepteur autorise son agent à s'y connecter pour ajouter les outils à son registre.
3. La description en langage naturel de l'outil, récupérée depuis le serveur, suffit à ce que l'agent comprenne son rôle et sache l'appeler si c'est pertinent.
4. S'il y a un appel, le MCP gère les échanges.

Exemple : via MCP, l'agent découvre qu'un outil de requêtage BigQuery est disponible. Il interroge donc BQ en langage naturel en envoyant la requête "donne-moi les ventes du T1", toujours via le MCP. Il reçoit (ventes_T1_2026) en retour, et le tout est loggé comme une partie de la conversation.

MCP est une bonne pratique pour donner une boite à outil facile d'accès à un agent conversationnel, pas forcément pour du traitement de masse. Des solutions à base d'API pure garantissent qu'il est possible de consommer beaucoup moins de tokens pour le même résultat.

Et la gouvernance, toujours la gouvernance

Maintenant que votre agent peut faire des bêtises, n'oubliez pas de le surveiller

Observabilité

La capacité de superviser, mesurer et comprendre le comportement interne d'un agent (quels outils il a appelés, pourquoi, combien de temps ça a pris)

...dont FinOps

Les LLMs et les appels APIs, ça coûte. Des dashboards de suivi existent avec des responsables à nommer.

indispensables en prod

Gestion des PII et des accès

Les PII (Personally Identifiable Information) doivent absolument rester protégées dans un contexte agentique. Les documents confidentiels ne doivent jamais fuiter.

Idempotence

C'est une garantie de sécurité essentielle qui assure qu'une action, même si elle est accidentellement répétée, ne causera jamais d'erreur ou de duplication.

votre responsabilité de design

Votre agent se repose sur ce que vous lui faites savoir

Attention côté gouvernance

Memory

L'expert de l'utilisateur

Savoir issu des interactions :
préférences, décisions, contexte utilisateur
La mémoire est construite par l'agent et isolée par utilisateur. Tout agent a un concept de mémoire, mais son design dépend du besoin.

RAG

L'expert des faits

Injection de savoir externe :
docs, wiki, bases de données
La source de vérité préexiste à l'agent. L'accès à ces connaissances est l'objet d'un choix de design. Un agent n'a pas forcément de RAG.

Le stockage d'informations spécifiques à l'utilisateur en mémoire expose à des fuites critiques.
Si les documents du RAG ne sont pas gouvernés, risque de baisse de qualité ET fuite d'information.

01

Faire de l’agentique en *2026*

Nos outils

Lesquels choisir pour votre projet ?

Le besoin	Le rôle	Exemples d'outils que nous utilisons
Modèles	Fournir la capacité de raisonnement, de planification et génération de contenu.	Propriétaire : Gemini (Google), Claude (Anthropic). Open Source : Mistral, Llama 3 (Meta)
Environnements de développement	Espaces de travail pour coder, expérimenter et entraîner les modèles.	Notebooks : Colab, Vertex AI Workbench IDE : VS Code. Assistés par IA : Cursor, Copilot, Gemini in VS Code, Claude Code
Frameworks d'orchestration	Bibliothèques pour construire la logique de l'agent (gestion des outils, mémoire, chaînage).	LangChain, LlamaIndex, CrewAI, Google Agent Development Kit (ADK), Gemini Enterprise for Customer Experience (GECX)
Prototypage no-code / low-code	Plateformes avec UI pour créer et tester rapidement des prototypes d'agents.	Gemini Enterprise (Agent Designer), Voiceflow, Relevance AI, N8N, Lovable
Ops : déploiement & supervision	Mise en production et supervision performance, qualité et coûts (FinOps).	Plateformes Cloud : Vertex AI (GCP), Azure Qualité & Debug : Giskard, LangSmith, Datadog

La structure des coûts

Entraînement, construction des LLMs

Inférence, run, utilisation des modèles

Infrastructure Cloud ou coût infra physique

Modèle	Entreprise	Coût API (Mo tokens)	Abonnement
GPT-4o	OpenAI	$2.5/M in, $10/M out	$0 à $200 par mois Business $20/emp.
GPT-4o Mini		$0.15/M in, $0.60/M out	
Gemini 3.1 Pro		$1/M in, $6/M out	$0 à $19.99 par mois Business de $21 à $70/emp.
Gemini 3.5 Flash		$1.5/M in, $9/M out	
Gemini 3.1 Flash Lite		$0.25/M in, $1.5/M out	
Claude 4.5 Haïku	ANTHROP\C	$1/M in, $5/M out	$0 à $100 par mois Business de $20 à $100/emp.
Claude 4.6 Sonnet		$3/M in, $15/M out	
Claude 4.7 Opus		$5/M in, $25/M out	
Mistral 8x22B		dépend de l'hébergement	$0 à $14,99 par mois Business custom
Mistral Large 2		$4/M in, $12/M out	
DeepSeek v4-Pro	deepseek	$0.0036/M in, $0.87/M out	_

Le coût des APIs est le gros de la dépense en production, même si le coût d'un projet dépend aussi de l'infrastructure.

L'Agentic Era au prisme de Google Cloud Next

Google Cloud Next '26, Las Vegas · 22–24 avril 2026

260
annonces produit, partenaires, clients

32 000
leaders & développeurs présents

~75 %
des clients GCP utilisent les produits AI

de 10 à 16 Md
tokens / minute traités via API au Q2

01 PLATEFORME

Gemini Enterprise Agent Platform

Poste de pilotage unifié pour **construire, scaler, gouverner & optimiser** les agents.

Agent Studio, Registry, Identity, Gateway, Observability
Orchestration A2A
Modèles : Gemini 3.1 Pro, Nano Banana 2, Lyria 3, Claude Opus 4.7

02 APP

Gemini Enterprise App

La porte d'entrée AI pour **chaque collaborateur**, du métier au tech.

Agent Designer no-code
Inbox de supervision
Long-running agents en sandbox sécurisée
Skills, Projects, Workspace Studio

03 INFRA

AI Hypercomputer & TPU 8

Une 8e génération de TPU en **dual-chip** pour absorber l'échelle agentique.

TPU 8t — training, performance triplée par rapport à la génération précédente
TPU 8i — inference, +80 % perf/$
Virgo Network & Axion N4A (Arm), GA

04 DATA

Agentic Data Cloud

Architecture data **AI-native, cross-cloud** pour donner aux agents la vitesse et le contexte.

Cross-cloud Lakehouse (AWS, Azure, on-prem)
Knowledge Catalog : graphe sémantique d'entreprise
Deep Research Agent
BigQuery Comments-to-SQL

05 SÉCURITÉ

Agentic Defense x Wiz

Threat Intelligence + SecOps + Wiz en une plateforme de **défense agentique.**

Triage Agent : 30 min → 60 s, 5 M+ alertes traitées
Dark Web Intelligence, Threat Hunting Agent
Wiz AI Application Protection (AI-APP)

De l'expérimentation à l'entreprise agentique en production, c'est maintenant.

OFFRES
Des missions **Agentic Data Cloud** et **data readiness,** le vrai goulot.

DELIVERY
Migration **Gemini Enterprise Agent Platform** (gouvernance, identité, observabilité).

ÉCOSYSTÈME
Fonds partenaires, Agent Marketplace, **coopérations** inter-entreprises...

Notre Lab IA en première ligne

Une centrale d'accélération agentique interne

Dans ce paysage changeant, nous catalysons **l'adoption** des produits agentiques et centralisons notre **innovation** au Lab IA.

01
Innovation
RECHERCHE · RUPTURE

Explorer les ruptures en équipe avant le marché

Trouver des solutions aux problèmes complexes des tech builders

Publier nos travaux et affirmer notre thought leadership

02
Produit
INDUSTRIALISATION

Maîtriser les plateformes et outils les plus récents

Industrialiser le delivery : repeatable, testable, scalable

Préparer des assets service + outil à disposition de nos clients

03
Go-to-market
CONVERSION · PREUVE

Permettre aux clients de se projeter avec des démonstrateurs concrets

Crédibiliser nos offres par la preuve

04
Transfo. interne
AGENTIFICATION

Agentifier nos process internes (staffing, CV, veille)

Augmenter chaque consultant avec des outils IA au quotidien

Être le premier utilisateur de nos propres produits

Nos convictions chez Converteo

- gouvernance
- design
- intégration

- **1. Cas d'usage et gouvernance :** Chaque projet doit partir d'un besoin métier clair et d'une gouvernance établissant les rôles et indicateurs de succès
- **2. Orchestration multi-agents :** Nous privilégions une équipe d'agents spécialistes régie par un agent orchestrateur plutôt qu'un agent ultra polyvalent et single-point-of-failure
- **3. Stratégie modèles :** Le bon modèle est celui qui répond bien au besoin tout en ayant un arbitrage systématique sur sa performance, son coût et la souveraineté des données
- **4. Contexte et mémoire :** Pour garantir la pertinence des actions, nous gérons la mémoire de l'agent (court / long terme) et son accès à l'information (RAG)
- **5. Tools et Intégrations :** La valeur de l'agent se distingue par son action. Garantir la fiabilité, la sécurité et la pertinence des outils / applications connectés est indispensable
- **6. Plateforme et SI :** L'architecture doit s'intégrer sans bouleversement au sein de l' écosystème en place avec un appui sur les plateformes cloud et datasets du client
- **7. Risques et conformités :** Conformité et sécurité sont des prérequis pour bâtir la confiance sur les outils IA plutôt qu'une contrainte
- **8. Supervision / AgentOps :** Le déploiement n'est pas l'étape finale, nous supervisons en continu pour garantir sa performance dans la durée

02

C’est quoi, être *GCP-First ?*

Notre expérience

#AI Engineering

TRANSFORMER L'EXPÉRIENCE CLIENT VIA L'IA

LACOSTE

Concevoir et mettre à l'échelle un chatbot conversationnel fluide personnalisé et contextuel qui magnifie chaque étape du parcours client

DÉPLOYER UN AGENT VOCAL DE SERVICE CLIENT

Sosh

Automatiser un service client entièrement vocal basé sur une IA voice-to-voice pour répondre aux besoins des clients

PRODUIRE DU CONTENU CRÉATIF AT SCALE

ACCOR

Concevoir et déployer une interface centralisée pour accélérer la production, l'adaptation et la localisation de contenu créatif.

DÉMOCRATISER L'ACCÈS AUX DONNÉES

L'ORÉAL

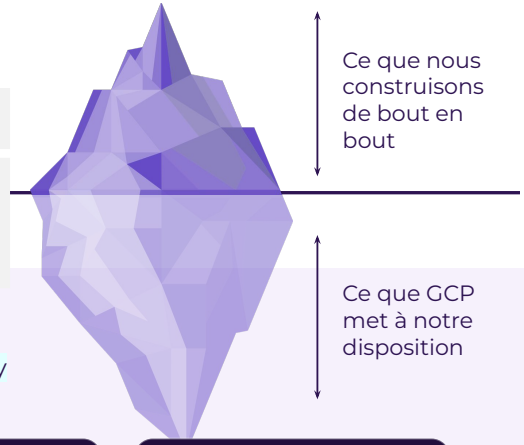
Déployer un agent conversationnel intelligent permettant aux équipes métiers d'obtenir des réponses immédiates et fiables à leurs questions, sans expertise technique.

L’infrastructure d’un projet

Tout pour l’échelle et l’accélération en production

Nous avons choisi de focaliser notre R&D sur l’écosystème GCP via un **partenariat majeur**.

En tant qu’early adopters, nous conseillons les clients, concevons **l’architecture** des agents et assurons leur industrialisation, tout **en réduisant les points de friction** grâce à l’expertise de Google. Nous menons ainsi chaque projet du POC au déploiement.



Gemini Enterprise
Gemini
Vertex AI
GCP Data Cloud
Looker
Gemini Enterprise for Customer Experience
ADK
BigQuery

Des produits performants

La pointe de la recherche de DeepMind sur les LLMs et l’IA, avec des mises à jour silencieuses

Une infrastructure managée

Maintenance facilitée et coûts réduits par le Cloud pour une grande quantité de données, produits off the shelf

Une offre complète

GCP, seul player à offrir compute, modèles avancés, plateformes agentiques et insights au même endroit

Une disponibilité maximale

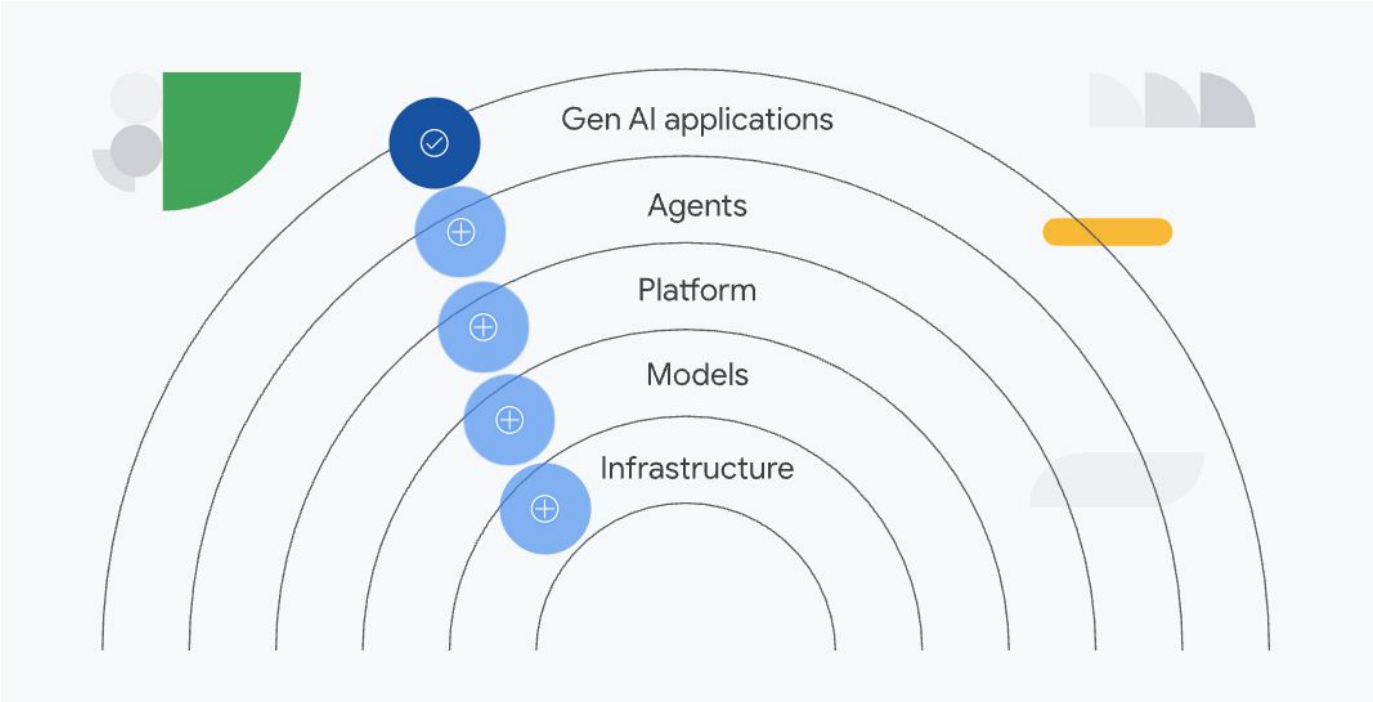
Modèles connectés interchangeables, machines virtuelles, calcul distribué, réseau Cloud mondial

Un support permanent

Des experts à l’ écoute des partenaires et des moyens financiers pour les projets les plus ambitieux

L’infrastructure d’un projet

Tout pour l’échelle et l’accélération en production

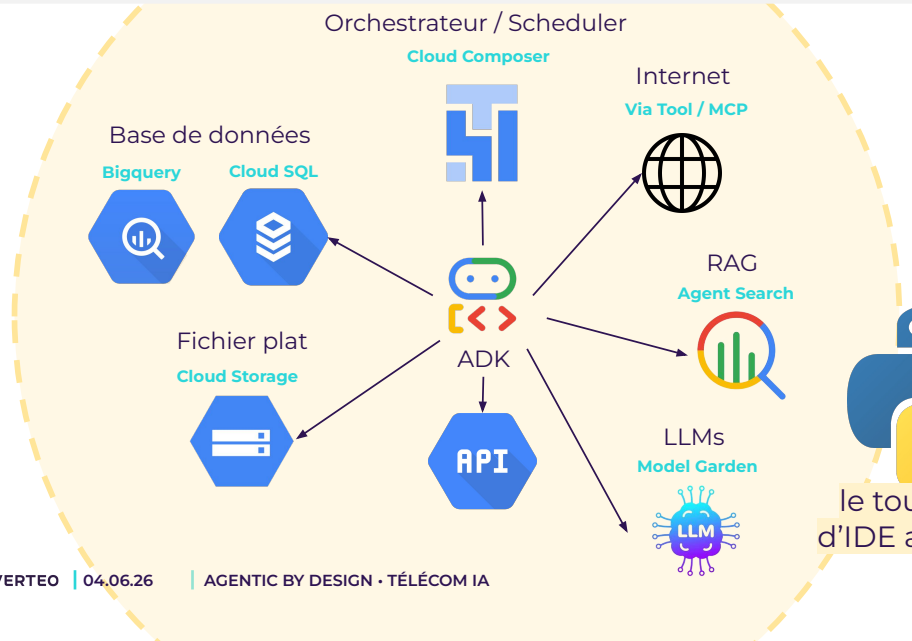


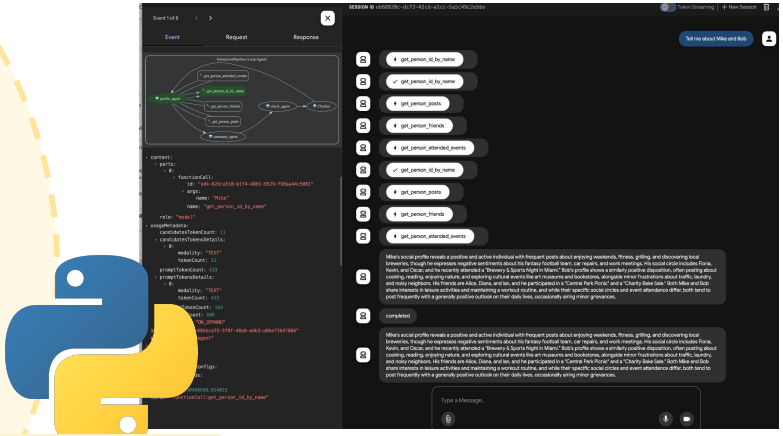
Agent Development Kit (ADK)

Un framework aux possibilités presque infinies

ADK est depuis 2025 le framework Google **le plus customisable** pour la livraison d'agents (callbacks, tools...).
Il permet de créer des setups multi-agents sans heurt et de se connecter à une variété de LLMs..

Il n'a pas d'UI. Il repose sur des compétences de dev : on code un agent en utilisant Python.
Sa force réside dans sa connexion facile avec les produits Google Cloud et des APIs tierces.



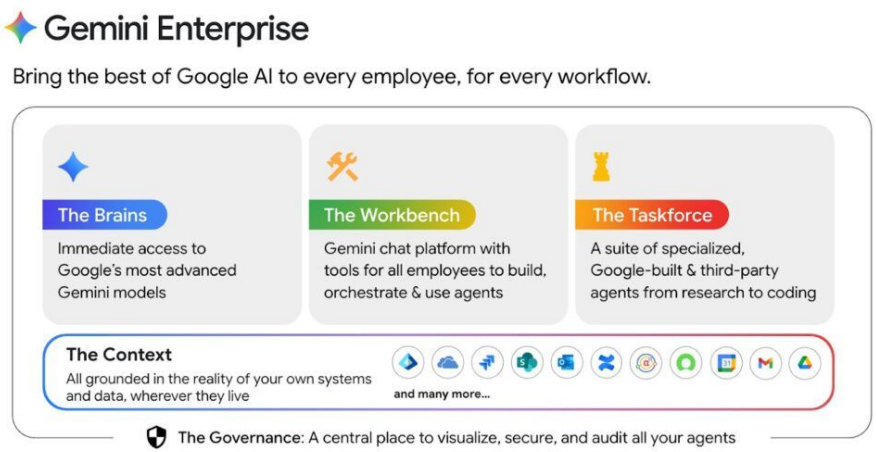


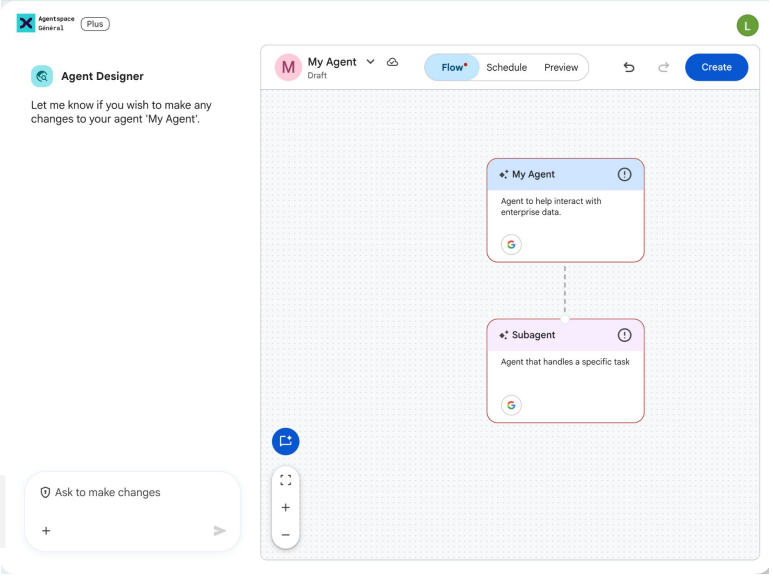
le tout à portée d'IDE avec Python

Gemini Enterprise

Une UI no-code pour tou-te-s les collaborateur-ice-s

Gemini Enterprise est une plateforme d'IA sécurisée pour les besoins de centralisation et les accès concurrents qui caractérisent une organisation. Elle s'est construite autour d'un chat Gemini augmenté par **des connecteurs sur les data sources d'entreprise** (dont third-party comme Slack ou Jira).





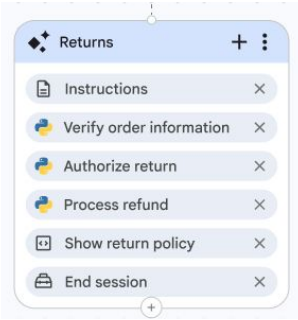
Elle propose une UI no-code pour créer **des agents individuels connectés à ces sources** et les déployer pour l'organisation.

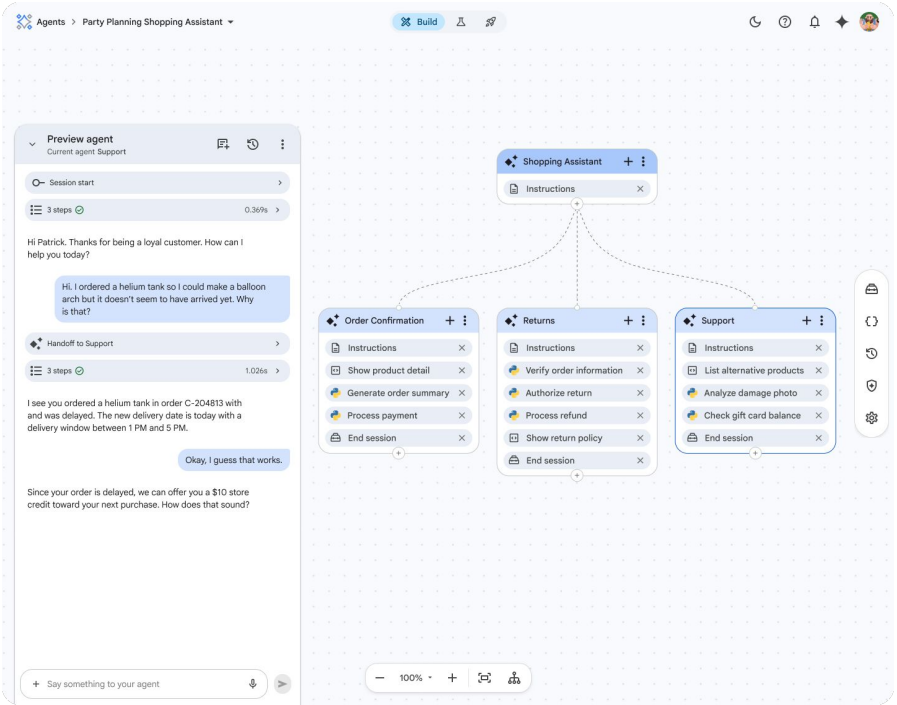
Gemini Enterprise for Customer Experience (GECX)

Le low-code pour le CCaaS Google (Contact Center as a Service)

GECX permet de développer chatbots et callbots à partir des derniers modèles Google sans s’embarrasser de l’infrastructure.

La solution comporte une UI low-code pour **créer et déployer des agents virtuels voix et texte** (CX Studio), un volet dédié au **support d’agents humains par IA** (Agent Assist) et une plateforme de **monitoring** (CX Insights).

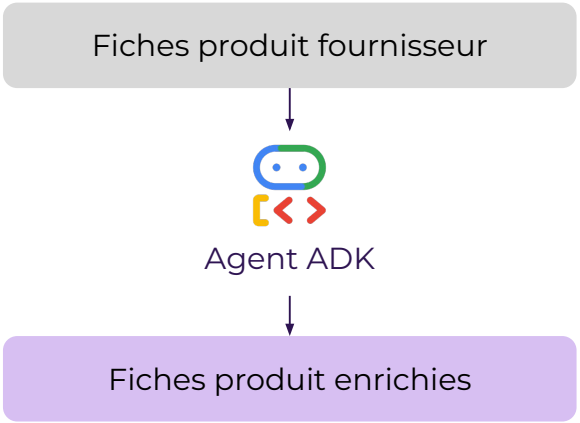




Une application chez Fnac Darty

Enrichir des fiches produit par une technologie agentique autonome

L’objectif est d’augmenter le nombre de ventes en améliorant la description, les caractéristiques, et les images des fiches produits. C’est un cas de traitement multimodal.



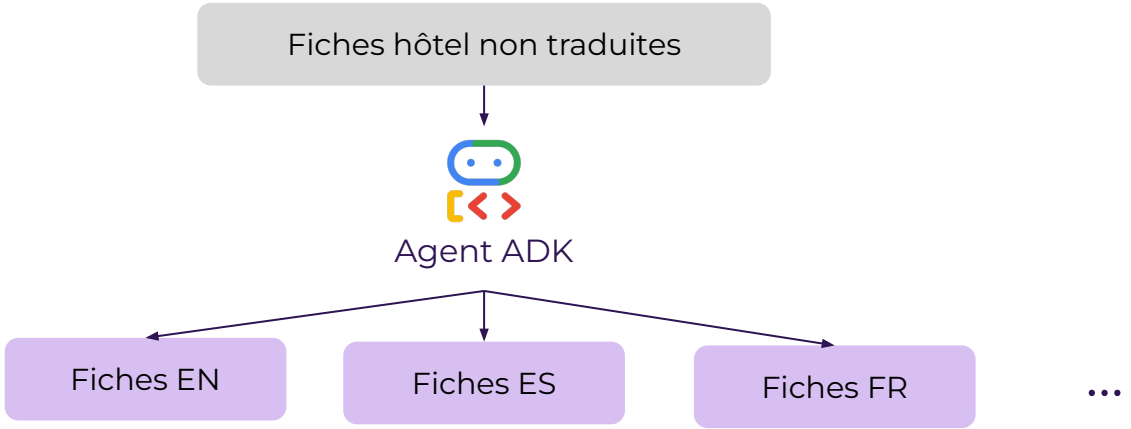
L’apport de Converteo ici semble être de penser l’architecture de l’agent avec ADK, puis de le déployer.
En réalité, c’est la préparation des données en amont et en aval qui est cruciale pour le succès du projet.

Une application chez Accor

Traduire des fiches d’hébergement en respectant l’identité de marque

Accor possède plusieurs hôtels dont les gérants poussent des descriptions des chambres dans leur langue locale.

Les traductions sont ensuites faites par une société externe qui facture à la fiche hôtel. L’objectif de ce projet est donc d'augmenter la vitesse de traduction, et de baisser les coûts liés aux prestataires qui traduisent les fiches.



Une application chez Adeo

Déployer un callbot voice-to-voice pour le service client avec GECX

Les callbots voice-to-voice commencent à apparaître chez les clients. Les premiers déploiements ont lieu ce Q2.
Le design du callbot par Converteo s’attache aux points suivants pour assurer la performance :

La transition vers le voice-to-voice

L’équipe a connu une approche textuelle auparavant, il faut changer la façon de penser les transcripts et les outils

Le ton

Le respect de l’image de marque appelle un flux déterministe souhaité par les métiers, mais nous devons préserver la performance

Le grounding

Nous ancrons l’agent dans les données d’entreprise (récupérer la FAQ, l’information client, et savoir comment les représenter)

La sécurité

La conformité RGPD et la confidentialité des données client dépendent de notre design de l’agent, et pas (seulement) de la plateforme logicielle

Le suivi qualité

Nous mettons en place des tests et utilisons des plateformes d’insights pour s’assurer de déployer un agent fiable.

L’accompagnement vers la production

En amont, la préparation des équipes et de l’infrastructure du client est cruciale pour le bon déroulement du projet et sa mise à l’échelle : bonnes pratiques, versioning & CI/CD, acculturation…

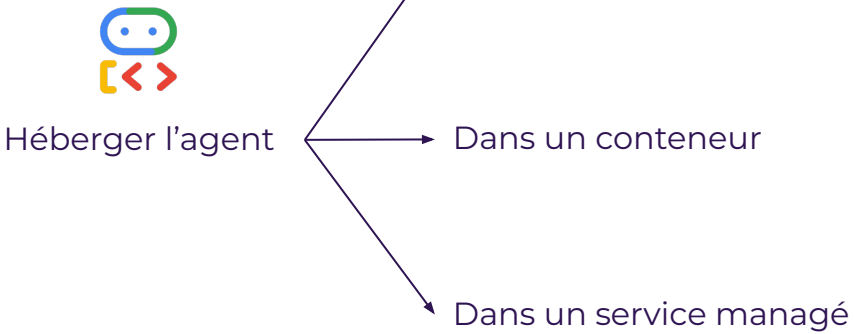
03

Et concrètement, comment *ça tourne ?*

Où habite votre agent ?

Concrètement, comment on héberge ça sur GCP ?

GCP offre trois options en fonction de l'utilisation des agents.



Compute Engine

Scaling vertical (approche classique), gestion des sessions à votre charge

Cloud Run

Scaling horizontal pour les charges très élevées, mais nécessite de penser l'architecture et de gérer les sessions

Agent Runtime

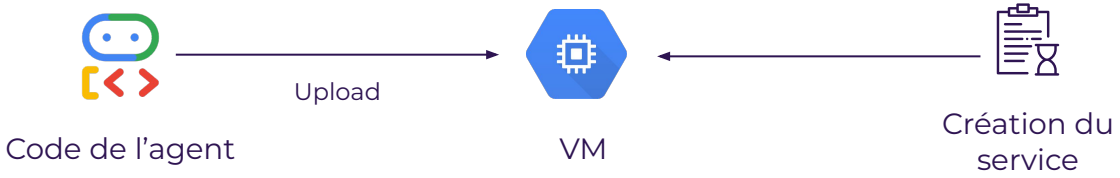
Zéro gestion d'architecture et sessions, mais limité à 90 appels/min/région — vite contraignant pour les traitements de masse

Les possibilités de GCP

Déploiement sur une VM

Compute Engine

Le développeur upload le code sur la VM, puis crée le service avec les bonnes règles d'exécution et des permissions fines via un compte de service. Il veille également à configurer les règles de pare-feu / réseau adaptées pour rendre son agent accessible, soit en interne, soit en externe dans le cas d'un déploiement public.



Limites :

- Scaling vertical (augmentation des perfs = redimensionnement de la VM avec plus de coeurs, plus de RAM...)
- Autoscaling difficile voire impossible à mettre en place
- Pas de véritable CI/CD même s'il est possible de conteneuriser

Les possibilités de GCP

Déploiement sur une VM

Compute Engine

```
my-adk-project/
├── .env                    # Clés API, variables
├── requirements.txt        # google-adk, dépendances
├── README.md
└── my_agent/
    ├── __init__.py         # from . import agent
    ├── agent.py            # définit root_agent
    ├── prompt.py           # instructions / system prompt
    ├── tools/
    │   ├── __init__.py
    │   └── custom_tools.py   # FunctionTool, etc.
    └── sub_agents/
        ├── __init__.py
        └── researcher/
            ├── __init__.py
            └── agent.py
```

```
# /etc/systemd/system/adk-agent.service

[Unit]
Description=ADK Agent API Server
After=network.target

[Service]
Type=simple User=adk
WorkingDirectory=/opt/my-adk-project
EnvironmentFile=/opt/my-adk-project/.env
ExecStart=/opt/my-adk-project/.venv/bin/adk api_server
--host 0.0.0.0 --port 8000
Restart=on-failure
RestartSec=5

[Install] WantedBy=multi-user.target
```

Recharger les services :

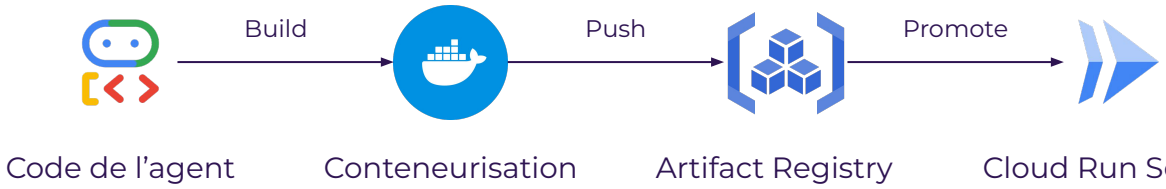
```
sudo systemctl daemon-reload
sudo systemctl enable --now adk-agent.service
```

Les possibilités de GCP

Déploiement en conteneurisant la solution

Cloud Run

Le développeur construit son image Docker et la pousse sur le registre (Artifact Registry), puis déploie le service Cloud Run avec les bonnes règles d'exécution et des permissions fines via un compte de service. Il veille également à configurer les règles d'accès réseau adaptées (ingress, IAM / authentification) pour rendre son agent accessible, soit en interne, soit en externe dans le cas d'un déploiement public.



Avantages :
- Scaling horizontal natif possible (multiplication des conteneurs)
- Portabilité du code
- Rollback possible (avec l'historique des images sur Artifact Registry)

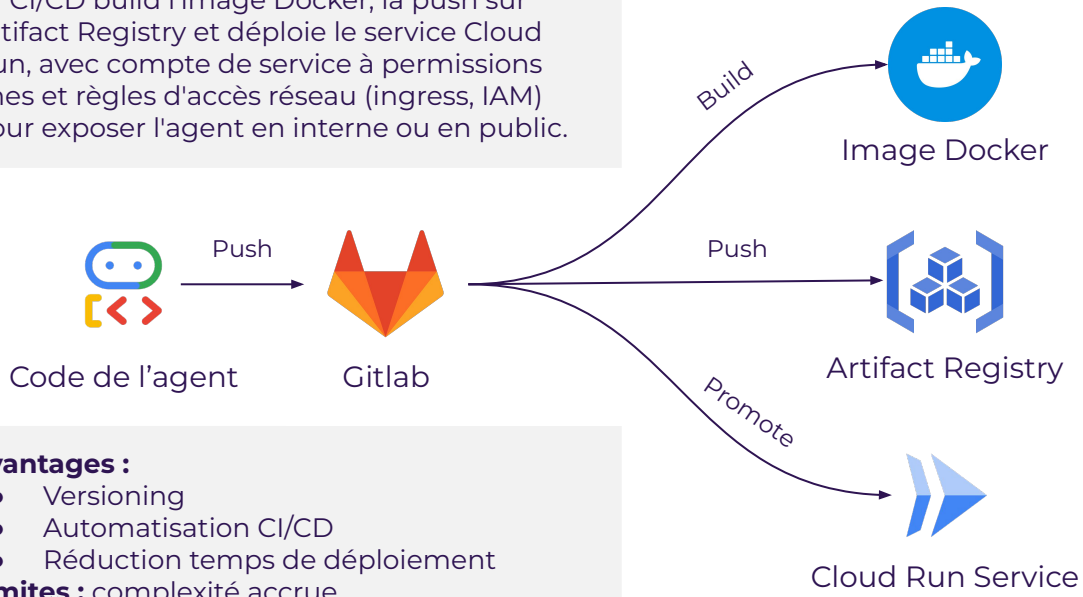
Limite : pas de CI/CD

Les possibilités de GCP

Déploiement en conteneurisant la solution

Cloud Run

Le dev pousse son code sur GitLab.
La CI/CD build l'image Docker, la push sur Artifact Registry et déploie le service Cloud Run, avec compte de service à permissions fines et règles d'accès réseau (ingress, IAM) pour exposer l'agent en interne ou en public.



Avantages :
- Versioning
- Automatisation CI/CD
- Réduction temps de déploiement

Limites : complexité accrue

Pipeline CI/CD

01 Création de l'image

02 Stockage dans registre

03 Promotion de l'image

Les possibilités de GCP

Déploiement en conteneurisant la solution

Cloud Run

```
my-adk-project/
├── .env                    # Clés API, variables
├── requirements.txt        # google-adk, dépendances
├── README.md
└── my_agent/
    ├── __init__.py         # from . import agent
    ├── agent.py            # définit root_agent
    ├── prompt.py           # instructions / system prompt
    ├── tools/
    │   ├── __init__.py
    │   └── custom_tools.py   # FunctionTool, etc.
    └── sub_agents/
        ├── __init__.py
        └── researcher/
            ├── __init__.py
            └── agent.py
```

```
# my-adk-project/Dockerfile

FROM python:3.12-slim
# Bonnes pratiques Python en conteneur
ENV PYTHONUNBUFFERED=1 \ PYTHONDONTWRITEBYTECODE=1

WORKDIR /app # Dépendances d'abord (optimise le cache)
COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt
# Code de l'agent
COPY . . # Utilisateur non-root (sécurité)

RUN useradd -m adk && chown -R adk:adk /app
USER adk

EXPOSE 8000 # Serveur API plutôt que `adk run`
(interactif, non adapté à un conteneur)
CMD ["adk", "api_server", "--host", "0.0.0.0",
"--port", "8000"]
```

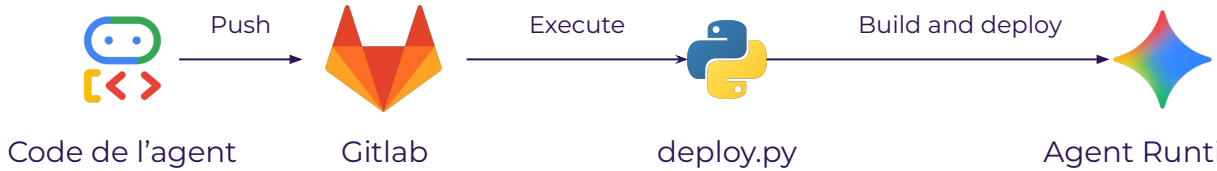
Next step : push de l'image sur Artifact Registry + déploiement Cloud Run
Remarque : on peut faire plus simple encore, et utiliser la commande `adk deploy cloud_run`

Les possibilités de GCP

Déploiement en conteneurisant la solution

Agent Runtime

Le dev pousse son code sur GitLab, puis la CI/CD exécute le script deploy.py qui package et déploie l'agent sur Agent Runtime (ex-Agent Engine), avec compte de service à permissions fines et règles d'accès réseau (IAM) pour exposer l'agent en interne ou en public.



Avantages :
- Scaling automatique
- Pas de complexité
- Logs et traces des temps d'exécutions + utilisations tokens natifs

Limite : Besoin d'un connecteur supplémentaire pour privatiser les connections

Les possibilités de GCP

Déploiement avec un script Python

Agent Runtime

```
my-adk-project/
├── .env
├── requirements.txt
├── README.md
└── my_agent/
    ├── __init__.py
    ├── agent.py
    ├── prompt.py
    ├── tools/
    │   ├── __init__.py
    │   └── custom_tools.py
    └── sub_agents/
        ├── __init__.py
        └── researcher/
            ├── __init__.py
            └── agent.py
```

```python
# my-adk-project/deploy.py
import vertexai
from my_agent.agent import root_agent # root_agent ADK

client =
vertexai.Client(project="mon-projet-gcp",location="us-central1")

remote_agent = client.agent_engines
              .create(agent=root_agent,config={
              "staging_bucket": "gs://mon-bucket-staging",
              "requirements":
              ["google-cloud-aiplatform[agent_engines,adk]"],
              "extra_packages": ["./my_agent"],
              "display_name": "Mon Agent",
        },
)
print(remote_agent.api_resource.name)
# note le resource_name complet
```

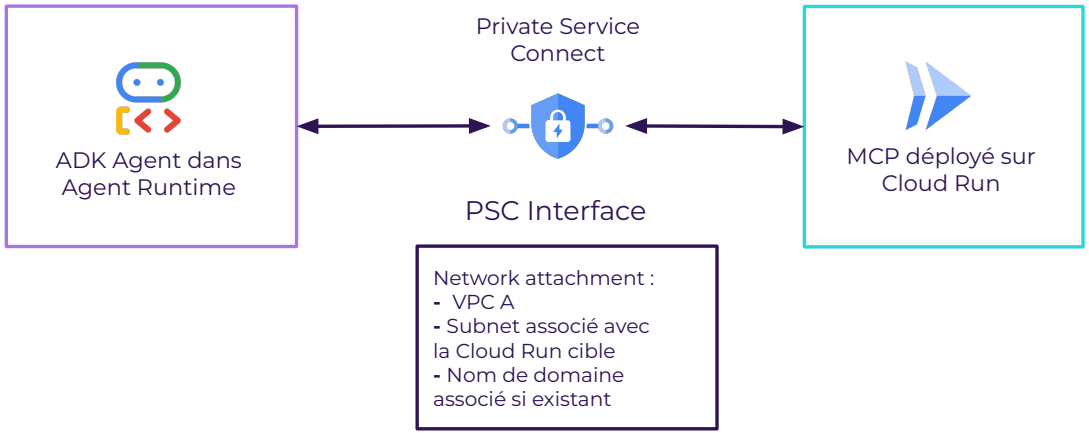
Remarque : Une méthode `update` existe pour mettre à jour l'agent depuis le script python. On peut aussi utiliser la commande CLI directement : `adk deploy agent_engine`, mais pas de mise à jour possible sans script python.

Pour aller plus loin

Privatisation des échanges

La **PSC** (Private Service Connect) **Interface** permet à l'agent (Runtime) d'accéder aux ressources privées du VPC. Elle crée un pont réseau privé entre l'agent de l'environnement managé Google et ton VPC. L'agent met ainsi une « patte » dans le réseau privé et peut joindre les ressources internes (bases de données, APIs, VMs...) sans passer par Internet, en trafic privé.

Exemple de communication avec MCP Interne



Prérequis rôles IAM du SA AI Platform
- Compute Network Admin
- DNS Peer

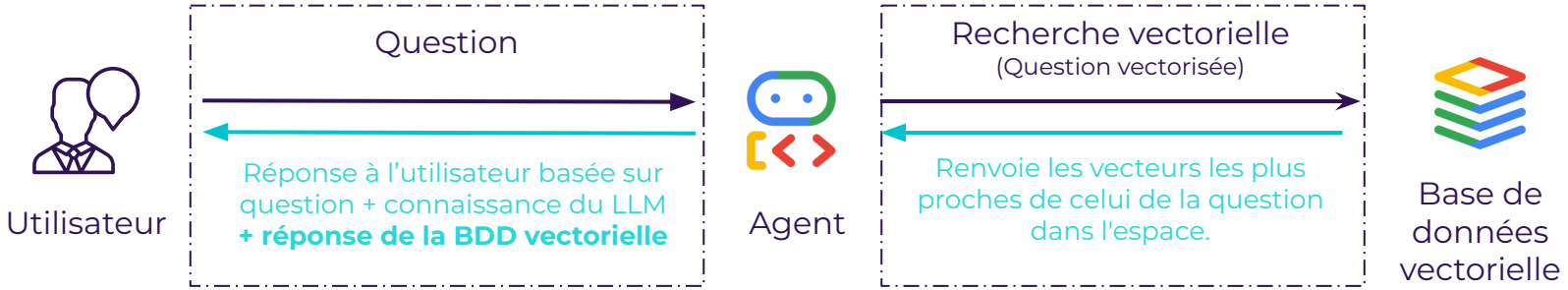
et pour binding sur le VPC A :
- Compute Subnetwork Use

Étapes :
1. Attribuer au SA AI Platform les rôles Compute Network Admin et DNS Peer
2. Sur le VPC A, binder le rôle Compute Subnetwork Use au SA AI Platform (accès au sous-réseau)
3. Créer la PSC Interface rattachée au sous-réseau du VPC A.
4. Déployer l'agent en référençant cette PSC Interface pour le routage vers le réseau privé.

Augmenter un LLM par du RAG

Qu'est ce que le RAG ?

Retrieve **A**nd **G**enerate

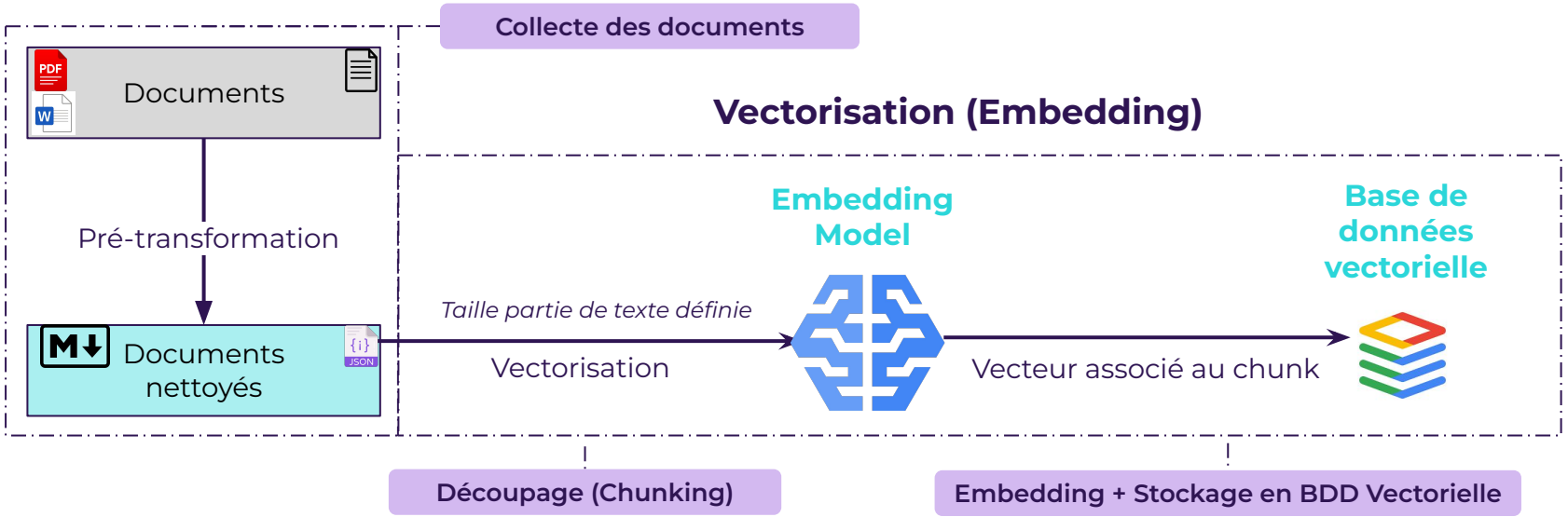


1 Question utilisateur comparée à la BDD

2 Résultat retourné dans le prompt + reformulation par l'IA

Alimenter une base vectorielle

Comment ça marche la base de données vectorielle ?



Proximité vectorielle ≈ proximité sémantique (le style et le vocabulaire comptent aussi)

En route pour la production : contrôle vs autonomie

Quel contrôle côté humain ? quelle autonomie pour cet agent ?

Les humains restent responsables quand il s'agit d'ajouter des **guardrails** (garde-fous) à l'agent,
de donner le **go / no-go** lors de sa mise en production ou d'une action critique,
ou enfin de **déléguer une tâche ou un workflow** dans son ensemble à un agent.

Privilégier le contrôle humain

Validation humaine à chaque étape ou avant une action critique
Agents utilisés pour préparer des ébauches de mails, synthèses etc..

Adapté quand :
risque élevé (juridique, financier..), faible maturité IA, forte sensibilité clients / marque

Augmenter l'autonomie de l'agent

L'agent peut enchaîner plusieurs actions sans validation car il a des garde-fous intégrés
Intégration profonde avec le SI (CRM, outils internes)

Adapté aux tâches :
bien cadrées, répétitives, fortement volumétriques (data quality, enrichissement..)

La conformité et l'éthique restent centrales.

En route pour la production : contrôle vs autonomie

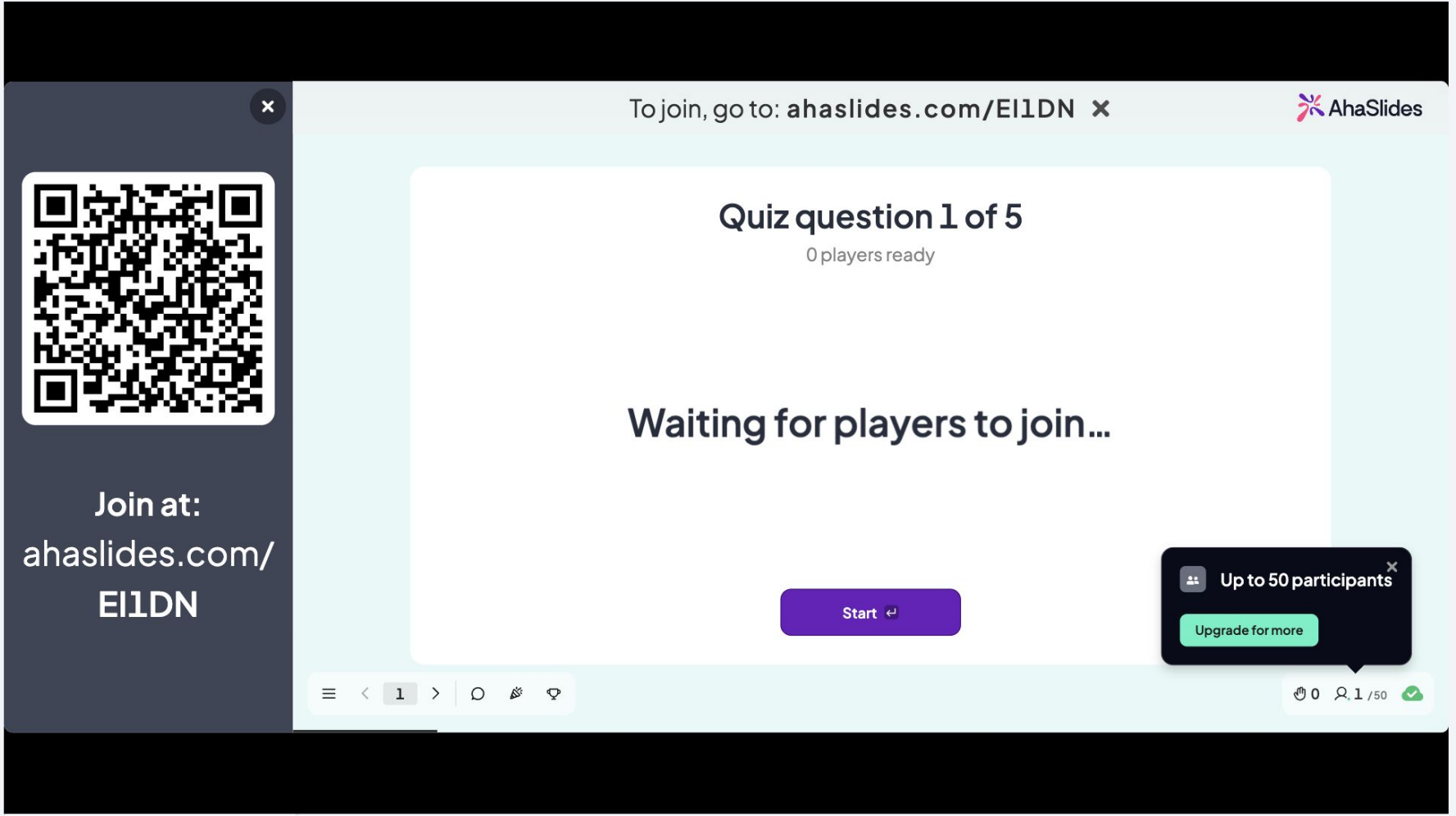
La posture Converteo dans un projet agentique

Pour travailler efficacement avec l'IA et les agents, voici la checklist 4D

Déléguer	Ce que je confie à l'IA : je clarifie sa tâche et son périmètre.
Décrire	Comment je donne le contexte : les bonnes informations, le processus, les limites.
Discerner	J'évalue sa réponse, sa cohérence, son alignement avec l'objectif.
Diligenter	Je reste responsable, documente, communique et tente d'améliorer la solution.

04

Petit *quizz*


To join, go to: ahaslides.com/El1DN
AhaSlides
Quiz question 1 of 5
0 players ready
Waiting for players to join...
Start
Up to 50 participants
Upgrade for more
Join at:
ahaslides.com/
El1DN

05

Du temps pour vos *questions !*

Motivé-e-s pour nous rejoindre ?

MEASUREMENT ENGINEERING

- Consultant Web Analytics

PLATFORM ENGINEERING

- Senior Full Stack dev
- Lead Full Stack dev
- Senior Data Engineer
- Senior Cloud Architect
- Lead Analytics Engineer

AI ENGINEERING

- **Stage AI Engineer**
- Senior AI Engineer
- Senior Lead IA Engineer

D’autres postes à venir,
dont des alternances.
Contactez-nous !

MERCI

leane.salais@converteo.com
brice.van-aken@converteo.com

Télécom IA · 04.06.2026