Module 1 · Session 1

The Digital Enterprise Context for AI

Digital foundations for AI-ready organizations

Leadership with AI – Cohort 2, IIT Bombay. Rajendra M Sonar, Ph.D.

Session Objectives

What participants should be able to do after this session

1. **Explain**
   Why AI cannot be implemented effectively in silos

2. **Describe**
   The broader digital context within which AI operates

3. **Identify**
   The role of data, cloud and connected systems in AI-led transformation

4. **Recognize**
   What makes an organization ready for AI adoption

Session Roadmap

1. **Digital primer for leaders**
   Common vocabulary for business and technology teams

2. **Why AI needs digital foundations**
   Start with the problem: AI value depends on data, systems and workflows

3. **Data management evolution**
   From files and databases to data lakes and lakehouses

4. **SQL, NoSQL, web and APIs**
   Core concepts that make digital systems work together

5. **AI, cloud, IoT and data (ACID)**
   How the key technologies combine in digital transformation

6. **AI-readiness discussion**
   Apply the concepts to the participant’s own organization

What technologies do?

A quick digital vocabulary for AI-ready leaders

Σ Compute

Process information
Run business logic

Provide UI/UX

Let users interact
Web and mobile screens

↔ Connect

Link systems, devices
and applications

✓ Access

Control permissions
Identity and security

Communicate

Enable messages
and collaboration

⇄ Exchange

Move data between
people and systems

DB Store

Keep records, files
and enterprise data

Automate

Execute workflows
with less manual effort

↓ Capture

Collect transactions,
sensor and user data

Manage Data

Clean, govern and
make data usable

Leadership takeaway: AI value depends on how well these basic digital capabilities are connected to business processes.

What managers should do about any technology?

A leader does not need to be a technologist, but must ask the right business, technology and adoption questions.

1 Business purpose

- Why is this technology needed?
- What is it used for?
- What value can be created or derived?

2 Technology understanding

- What does it do?
- What tools, techniques or methods implement it?
- How mature and reliable is it?

3 Enterprise fit

- How does it fit the task at hand?
- How will it integrate with existing systems?
- What data, APIs and infrastructure are needed?

4 Adoption readiness

- What skill-sets are required?
- What leadership and governance are needed?
- Who will own and operate it?

5 Risks and challenges

- What issues, limitations and risks exist?
- What are privacy, security and compliance concerns?
- What change-management barriers may arise?

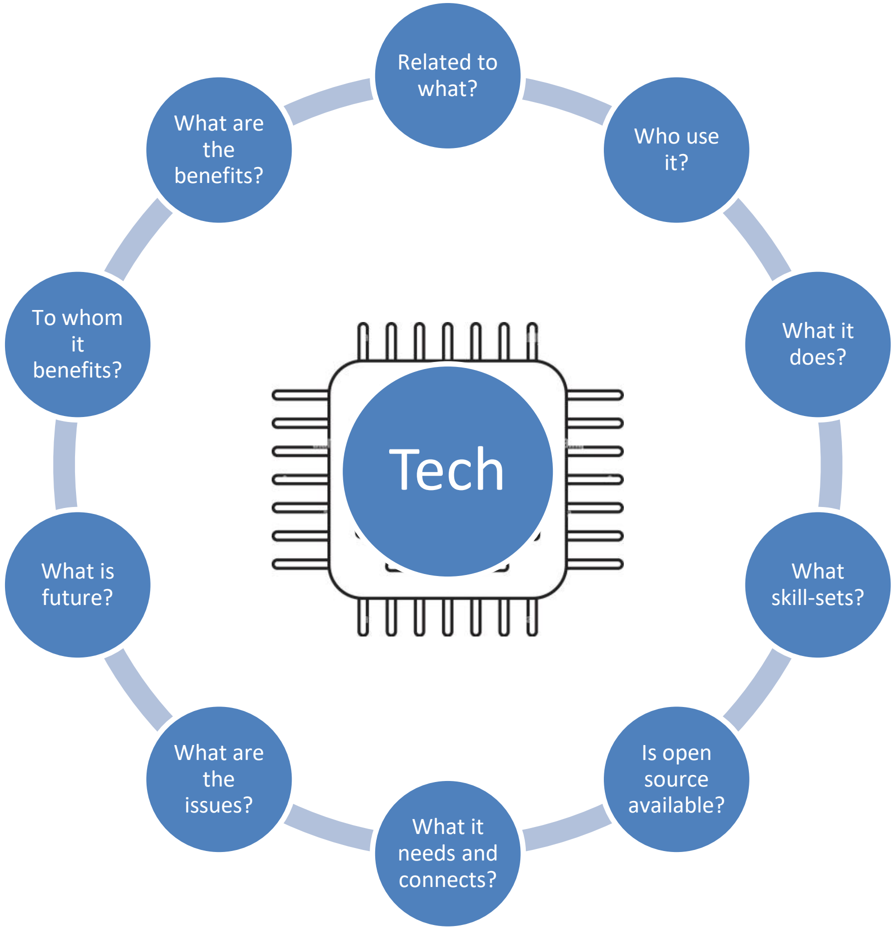
6 Stakeholders and impact

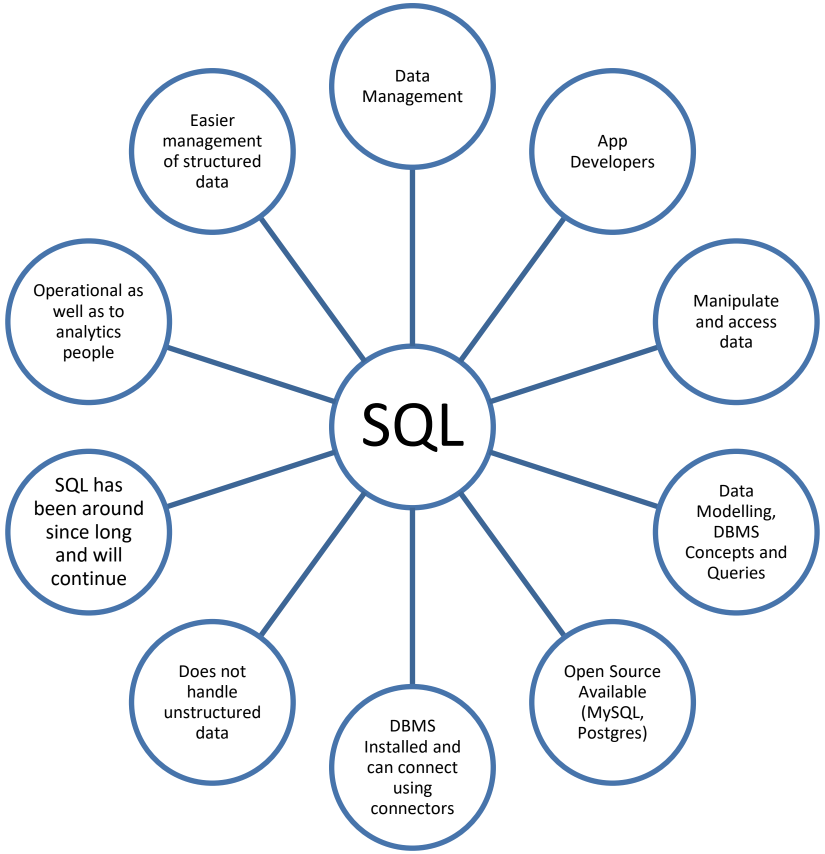
- Which stakeholders will use or be affected?
- How will it change workflows and decisions?
- How will success be measured?

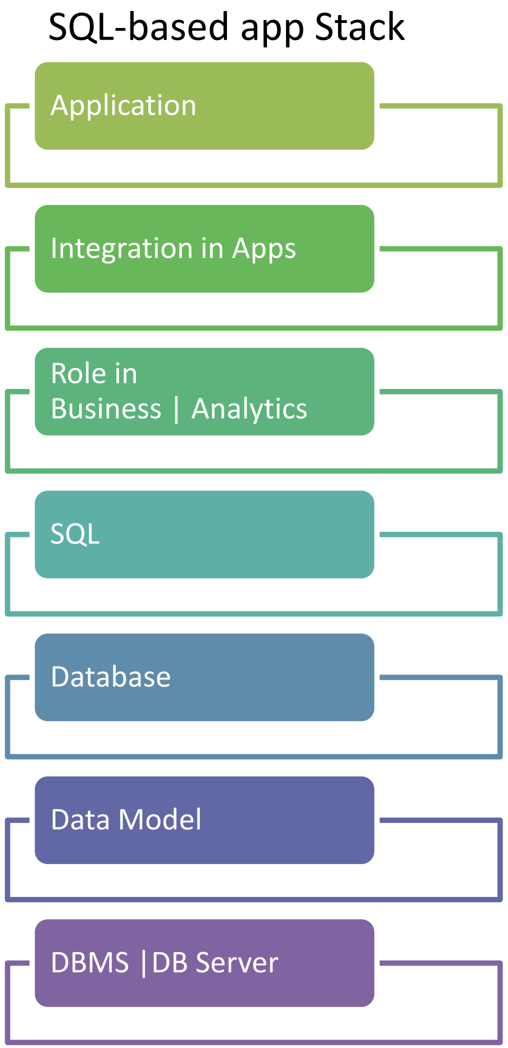
Better practice: create a technology portfolio/database

Capture: technology name, purpose, use cases, tools, owners, integrations, maturity, skill needs, risks, stakeholders and value metrics.

Understanding tech from managerial perspective?



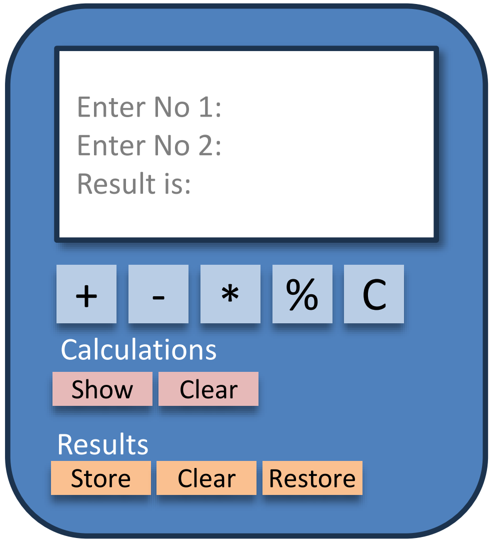




Understand tech well from managerial perspective, how they connect and fit into your existing stack….

Let us understand application/software components/architecture

A simple program to get two numbers and do math operations



1. Ask number 1
2. Validate Number 1 between 0-100
3. Ask number 2
4. Validate Number 2 between 0-100
5. User selects operation
6. Based on operation, calculations are done
7. Result is displayed
8. Result is automatically stored in memory
9. Calculations Button operations: **Show** shows calculations in memory, **Clear** clears memory
10. Results buttons: **Store** stores all calculations in database, C **Clear** from database, **Restore** brings calculations from database

1. Ask number 1
2. Validate Number 1 between 0-100
3. Ask number 2
4. Validate Number 2 between 0-100
5. User selects operation

6. Based on operation, calculations are done

6.1 Addition
6.2 Subtraction
6.3 Multiplication
6.4 Division

7. Result is displayed

8. Result is automatically stored in **memory**

9. Calculations Button operations: Show shows calculations in memory, Clear clears memory

10. Results buttons: **Store** stores all calculations in database, C **Clear** from database, **Restore** brings calculations from database

UI/UX

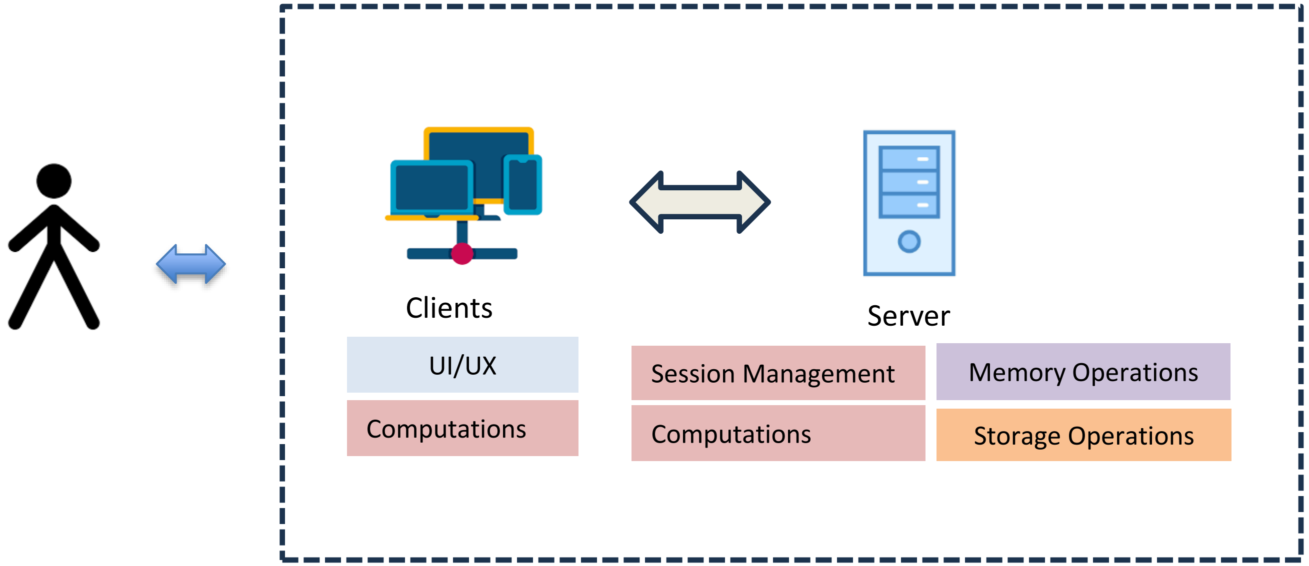
Computations

Memory Operations

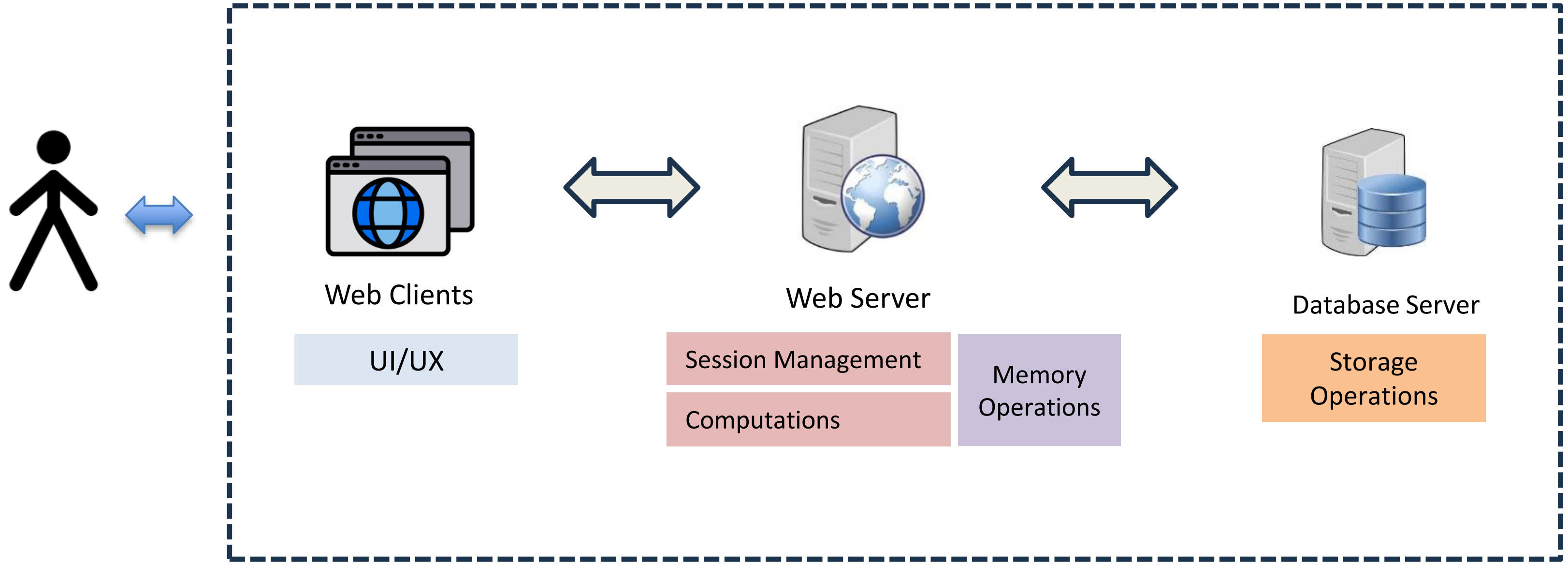
Storage Operations

You can have choice of programming language to use and split business logic

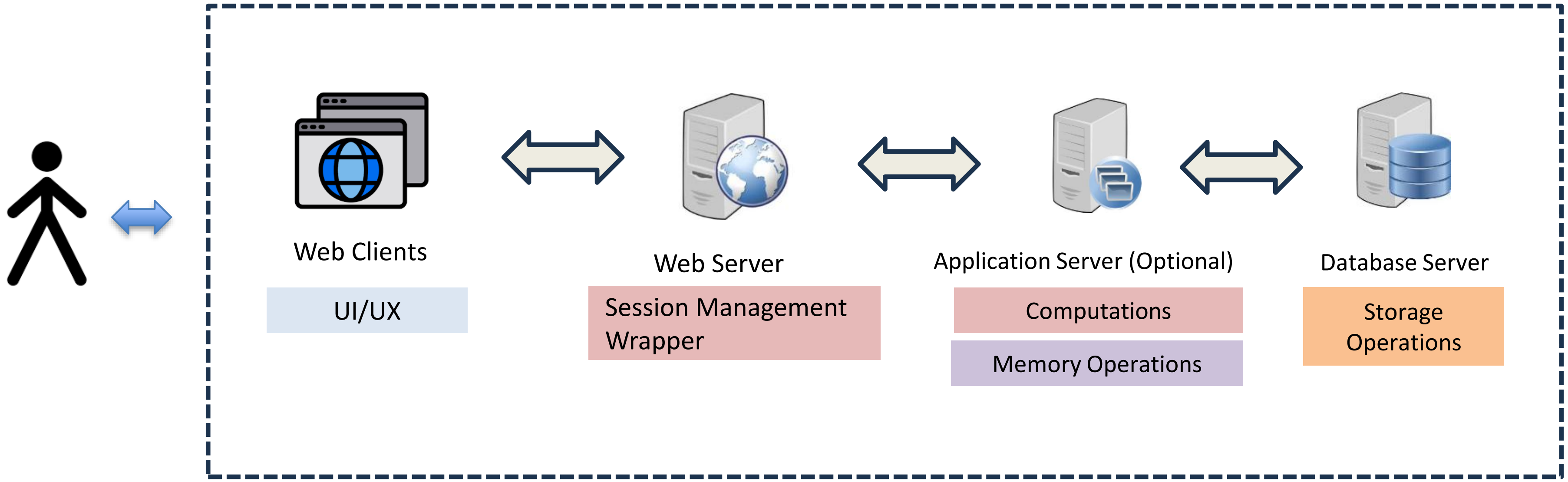
Let us understand application/software components/architecture



Let us understand application/software components/architecture

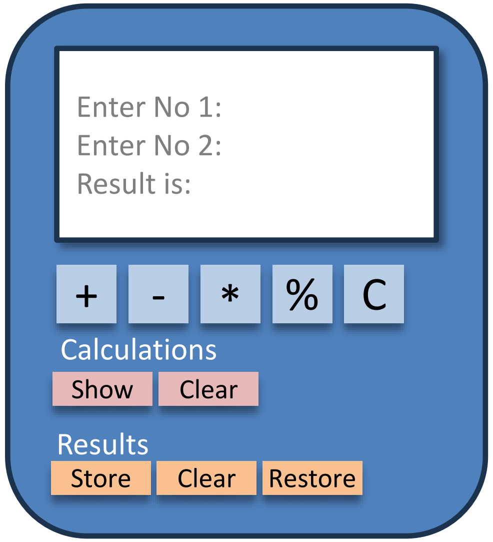


Let us understand application/software components/architecture



AI/ML Tasks, GPU servers, LLMs and agentic AI will change the architectures

Let us understand application/software components/architecture



1. Ask number 1
2. Validate Number 1 between 0-100
3. Ask number 2
4. Validate Number 2 between 0-100
5. User selects operation

6. Based on operation, calculations are done

6.1 Addition
6.2 Subtraction
6.3 Multiplication
6.4 Division

APIs

Client can access API

Common API
Separate APIs

7. Result is displayed

8. Result is automatically stored in **memory**

9. Calculations Button operations: Show shows calculations in memory, Clear clears memory

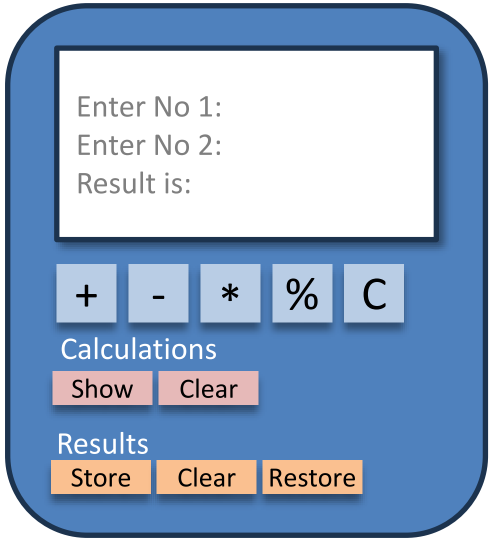
10. Results buttons: **Store** stores all calculations in database, C **Clear** from database, **Restore** brings calculations from database

DB APIs

Common API
Separate APIs

Let us understand application/software components/architecture

1. Ask number 1
2. Validate Number 1 between 0-100
3. Ask number 2
4. Validate Number 2 between 0-100
5. User selects operation



+ Microservice

1. Validate numbers
2. Add Operation

3. Result is automatically stored in **memory**

4. Return result

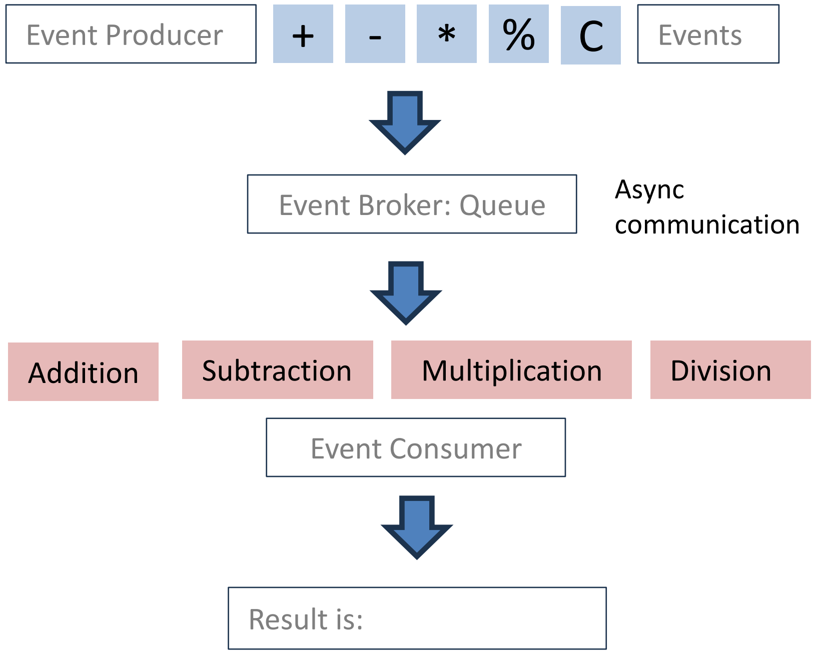
- Microservice

1. Validate numbers
2. Sub operation

3. Result is automatically stored in **memory**

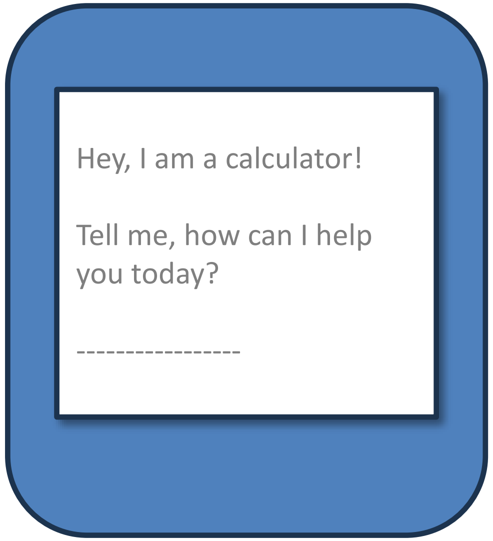
4. Return result

A **microservice** is an architectural approach to building applications as a collection of small, independent, and self-contained services. While an **API** (Application Programming Interface) is a mechanism or interface that enables different software components to communicate with each other.



Event-Driven Architecture (EDA) is a software design pattern where components communicate by producing, detecting, and reacting to **events** (significant state changes like a user action or data update), enabling decoupled, scalable, and responsive systems, common in microservices, using producers, consumers, and event channels (brokers) to facilitate real-time state changes across distributed applications

Let us understand application/software components/architecture



Think over, what does it change?

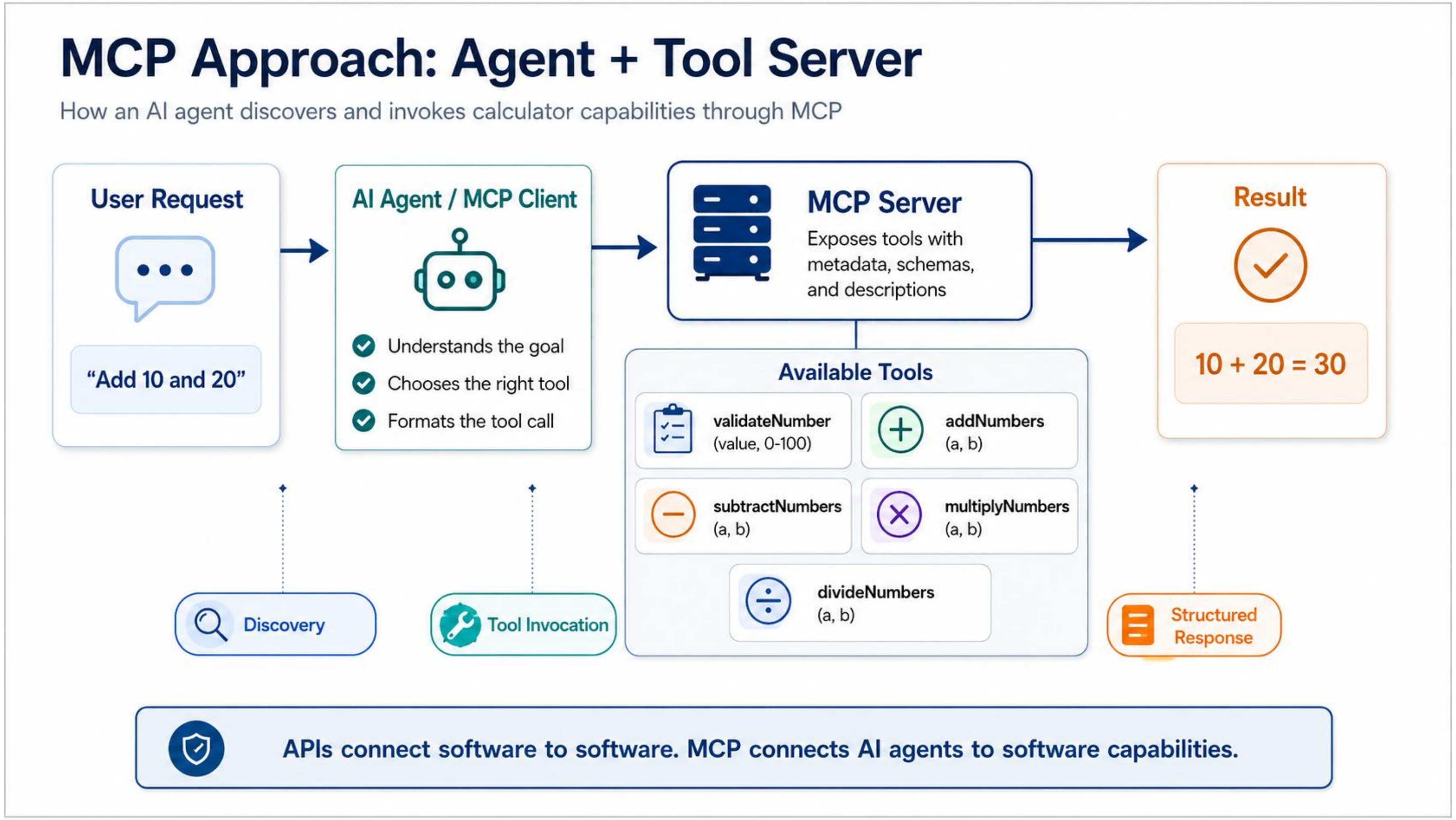
- Who is there?
- What the user is looking for?
- Let me find more about and recollect past experiences of the user?
- Is the user authorized and making valid request/task?
- What I need to do to fulfil the request, who will fulfil the request?
- I am done! Let me check validity of my response
- Let me send response to the user

Change in UI/UX (Multi-modal)	AI: Prompt and context engineering	Workflow change, automation of cognitive activities (deterministic-probabilistic)
Making services available through APIs and tools	Orchestrating everything for end-to-end automation	Human in loop and Guardrails

Let us understand application/software components/architecture

Hey, I am a calculator!

Tell me, how can I help you today?



Let us understand application/software components/architecture

- Server based
- PC based
- Monolithic
- Client/Server
- Web-based
- Three-tier
- Multi-tier
- Microservices
- API Driven
- Event Driven
- Agentic

Physical Server → Virtual Machine → Containers → Serverless

- Programming Language
- Frameworks
- APIs

- Software Development Shift
- SDLC!
- Waterfall/Agile

Less Infrastructure Management	Automatic Scaling
Efficient use of resources: Cost Efficiency	Focus on Development (Faster)

Open and accessible	Distributed
Open standard	Easily Integrable

Brings in lot of security and maintenance challenges

Computer Centric → Networked Centric → Human Centric → Device Centric

Architectural Shifts in Agentic AI

From traditional integration thinking to AI-agent-ready digital architecture

1

Deterministic → Probabilistic

Components that reason, not just execute predefined logic

2

API Contracts → Capability Discovery

Agents advertise capabilities via Agent Cards; dynamic binding

3

REST/gRPC → MCP + A2A + ACP

New protocol stack: agent-to-tool (MCP), agent-to-agent (A2A)

4

Orchestration → Goal-Oriented Autonomy

From predefined workflows to agents that plan, act, and reflect

5

External State → Memory as Architecture

Short-term, long-term, and shared memory are first-class concerns

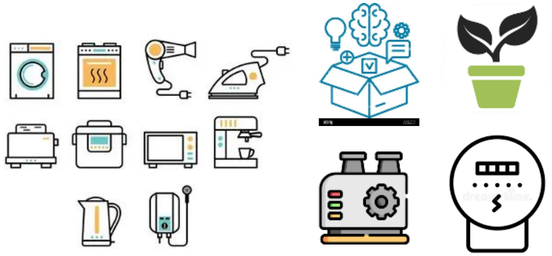
6

Event-Driven → Even More Critical

Loose coupling for agents; same lesson from early microservices

Internet of things

the things?



what makes the things **live**?

where are things?

the internet?

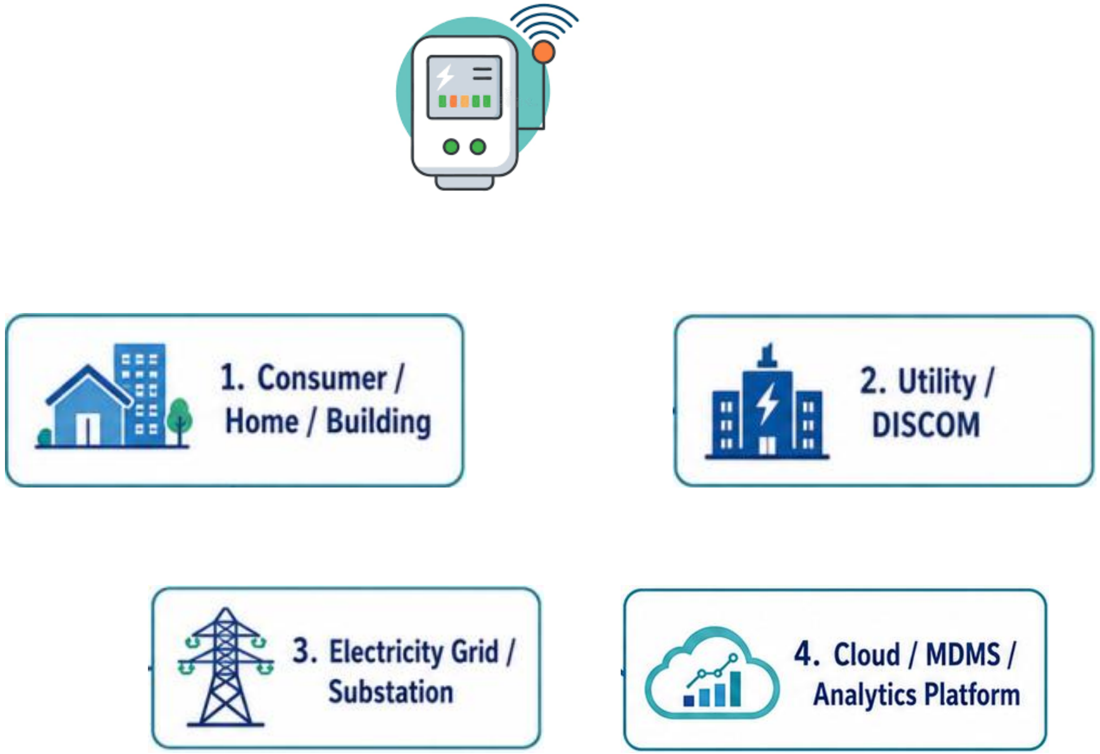
connecting to the internet?

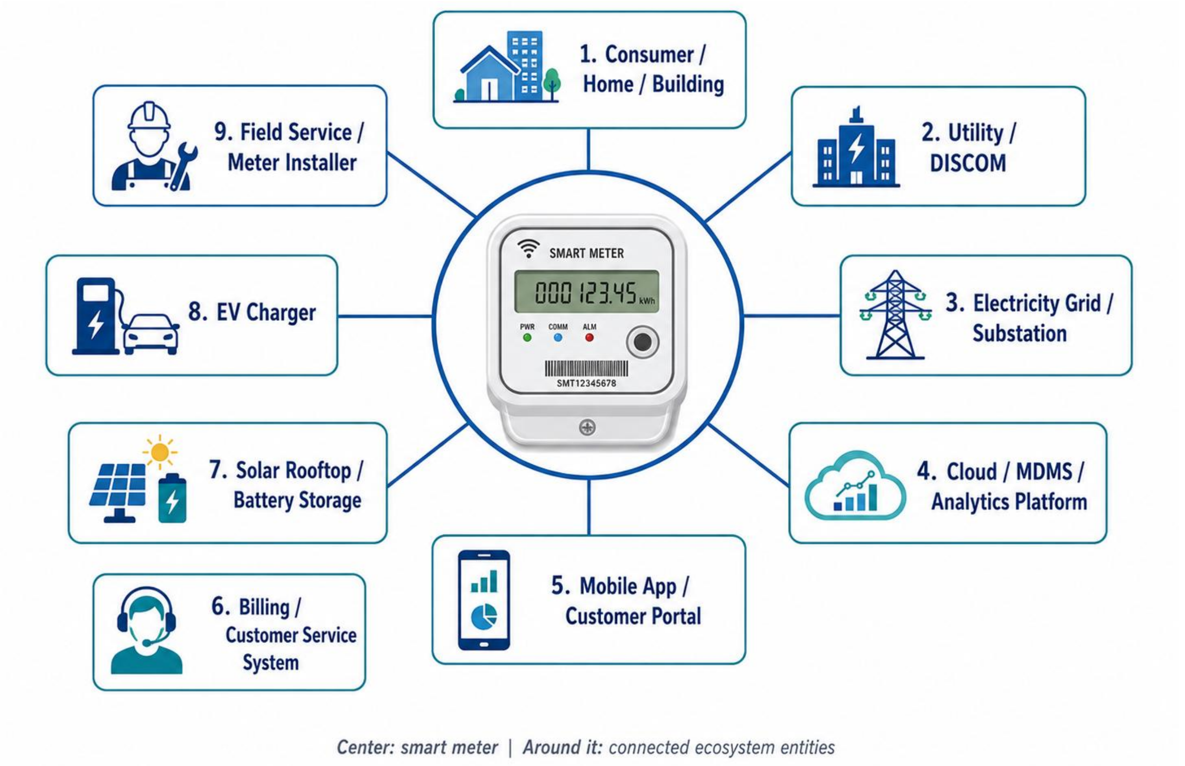
Creates an ecosystem, players and new business models

A complete digital and data driven ecosystem including digital twins

Internet of things

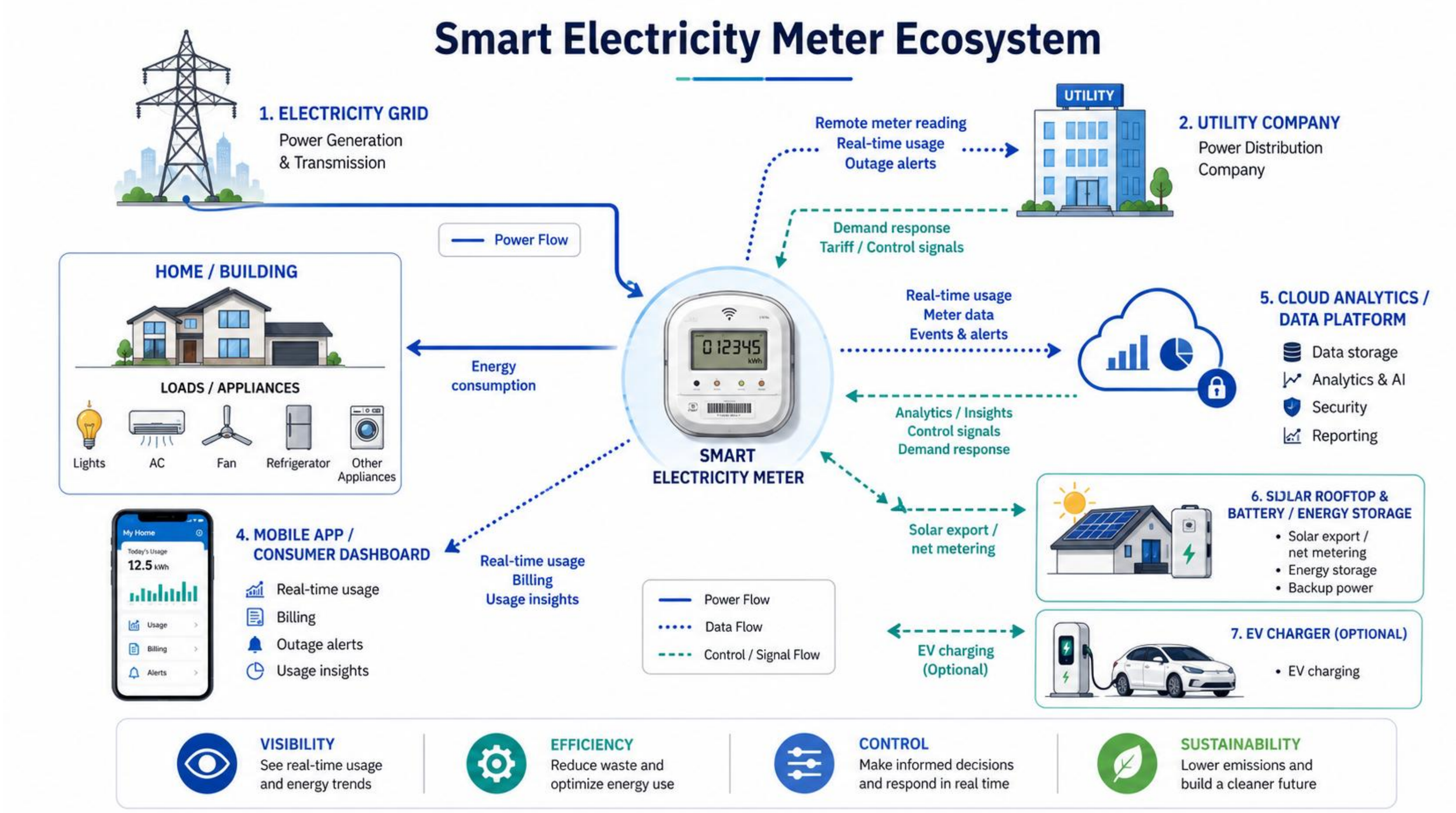
What ecosystem a smart meter can power?



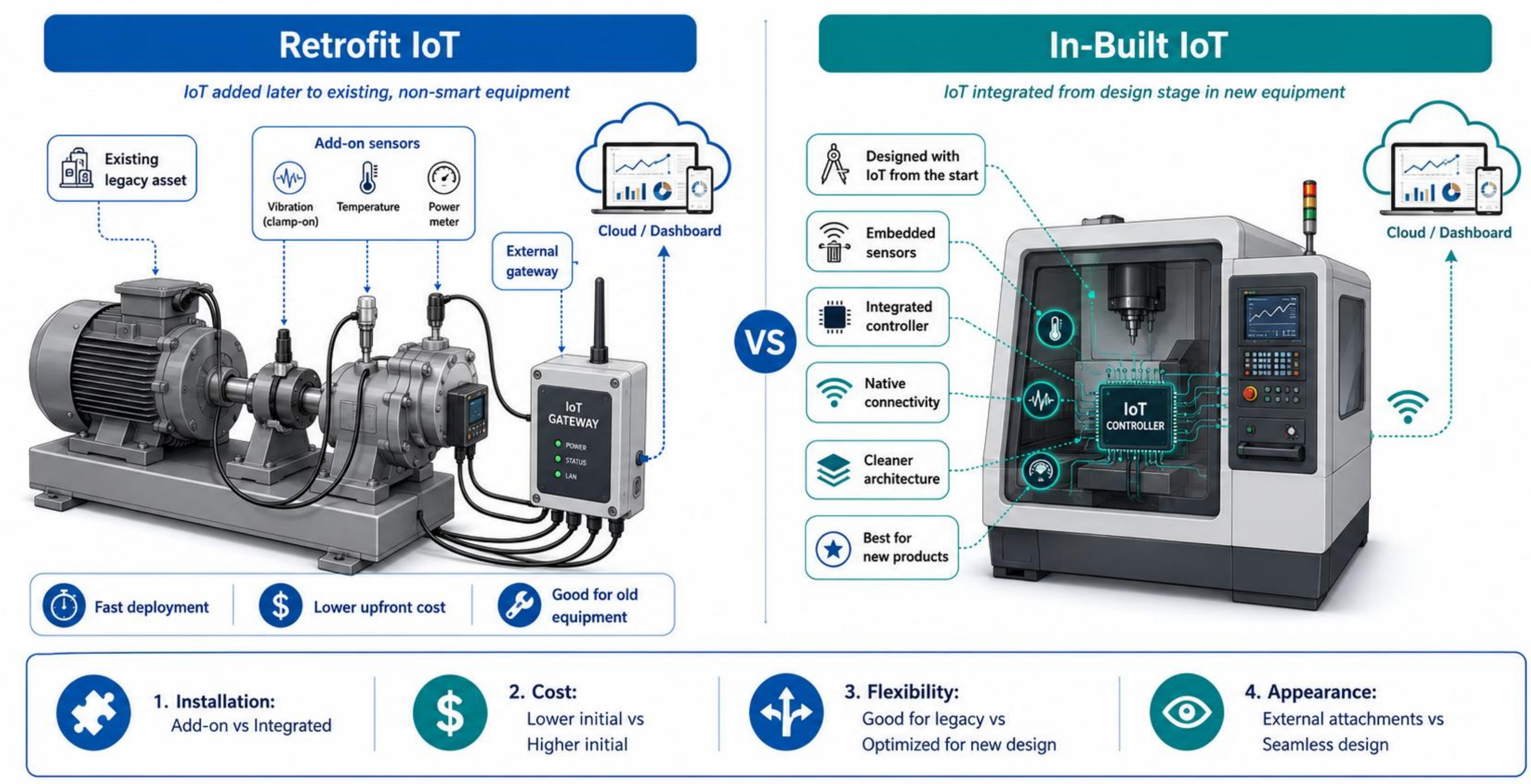


Think about CNG stations with IoTs?

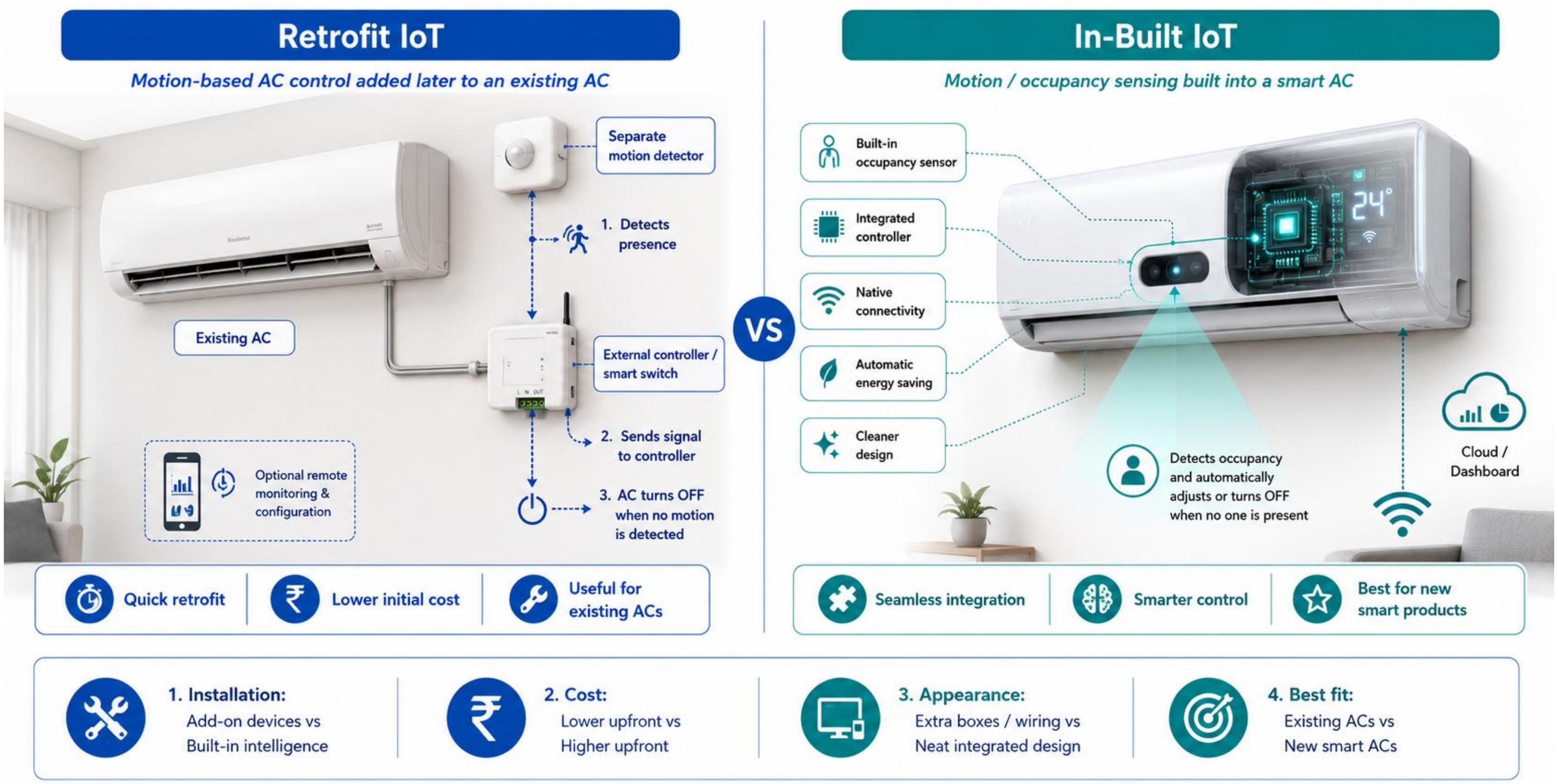
Internet of things



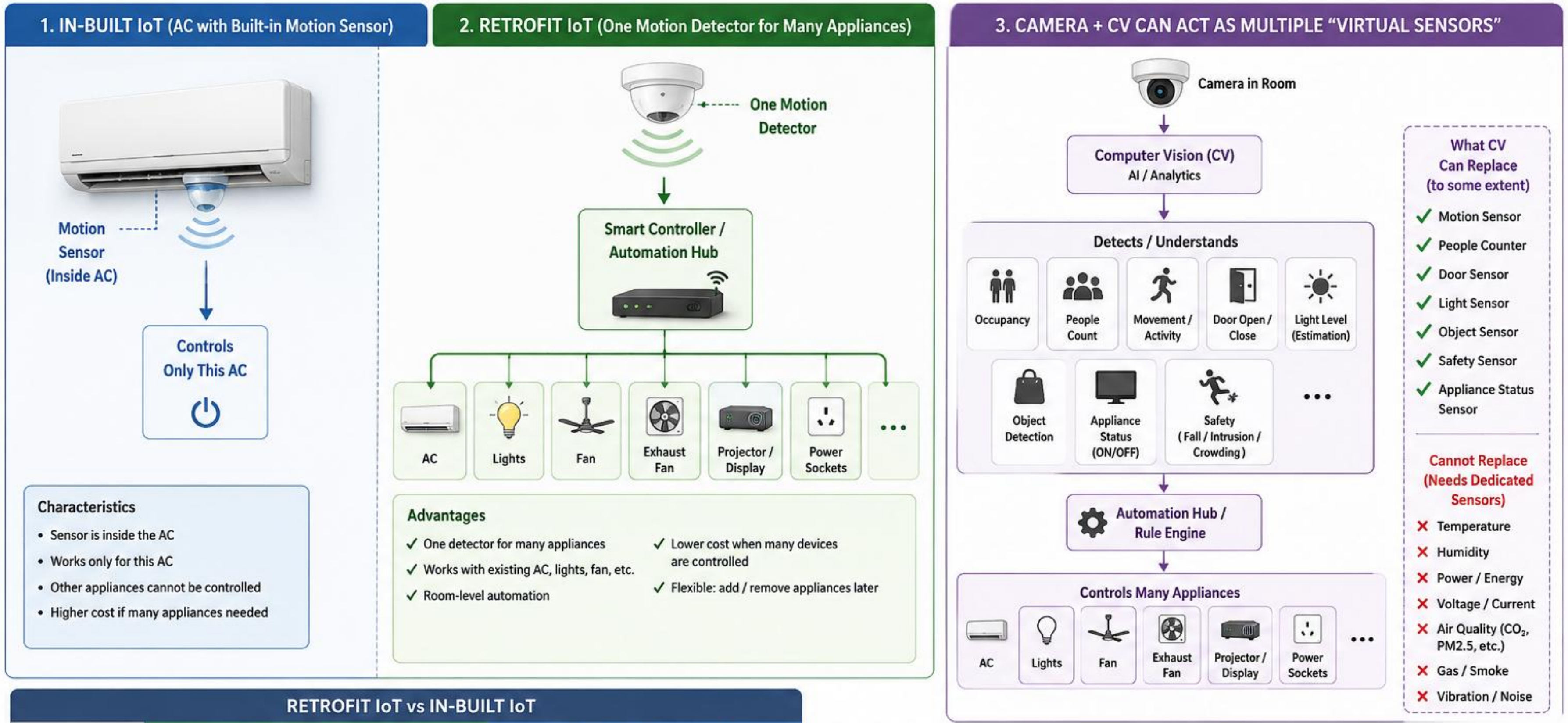
Internet of things



Internet of things



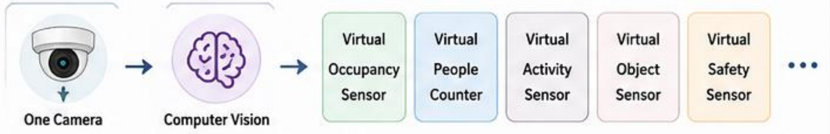
Internet of things



RETROFIT IoT vs IN-BUILT IoT

Aspect	Retrofit IoT (Room-level)	In-Built IoT (Product-level)
Scope	Controls many existing appliances in a room	Optimized for one appliance
Example	One motion detector controls AC + lights + fan + projector + sockets	AC has its own built-in motion sensor
Advantage	One sensor/controller for many devices, lower cost, flexible, easy to expand	Clean integratıler, better user experience, product-optimized
Limitation	Extra device(s), wiring/configuration, may be less integrated	Limited to that appliance, not reusable for others
Flexibility	Controls many existing devices	Optimized for one built-in product
Best Use	Homes, classrooms, offices, meeting rooms, hotels, retrofit projects	New appliances, premium products, single-device optimization

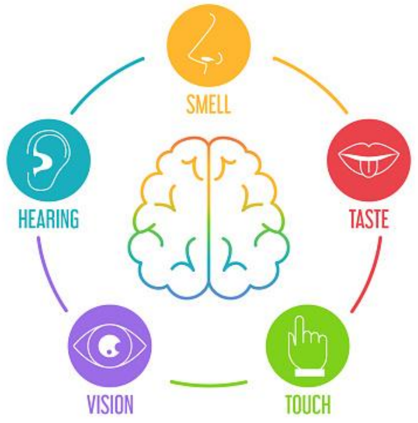
CAMERA + CV: FROM ONE CAMERA TO MANY VIRTUAL SENSORS

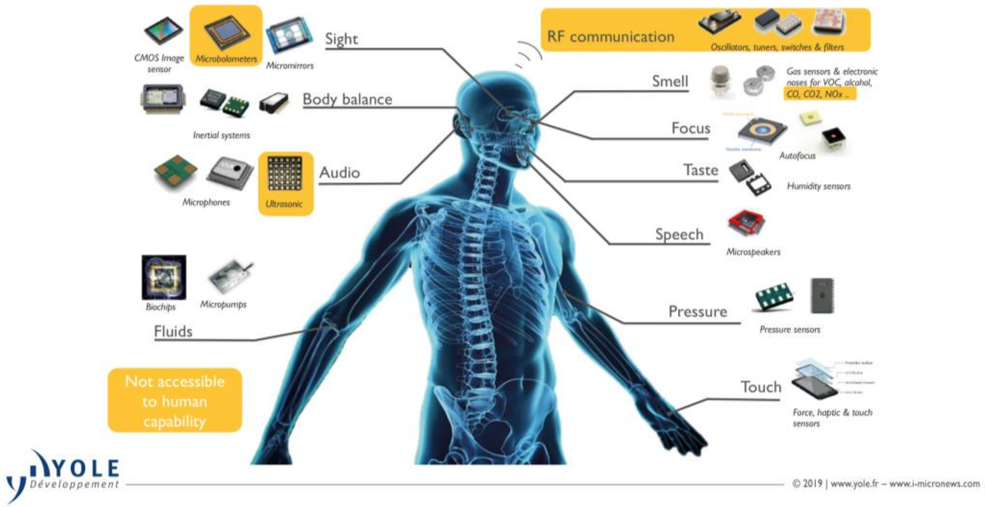


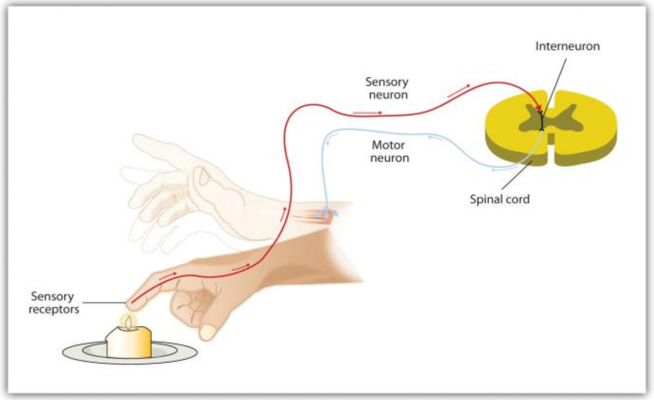
KEY TAKEAWAY

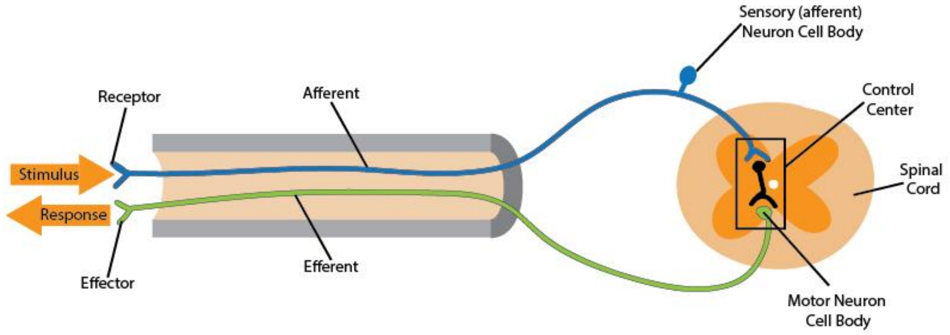
Camera + CV can substitute many presence, activity, counting, and visual-status sensors, making retrofit IoT more powerful and flexible — but physical measurement sensors are still needed for accurate environmental and electrical data.

Internet of things

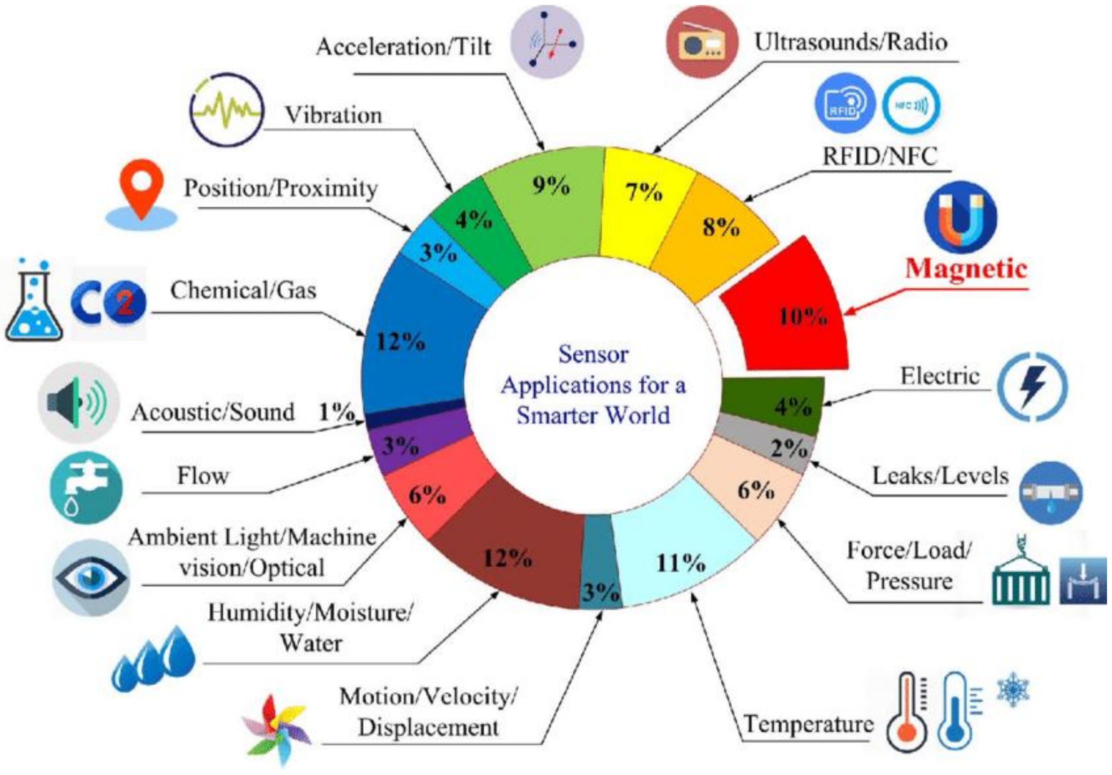








Internet of things



Internet of things

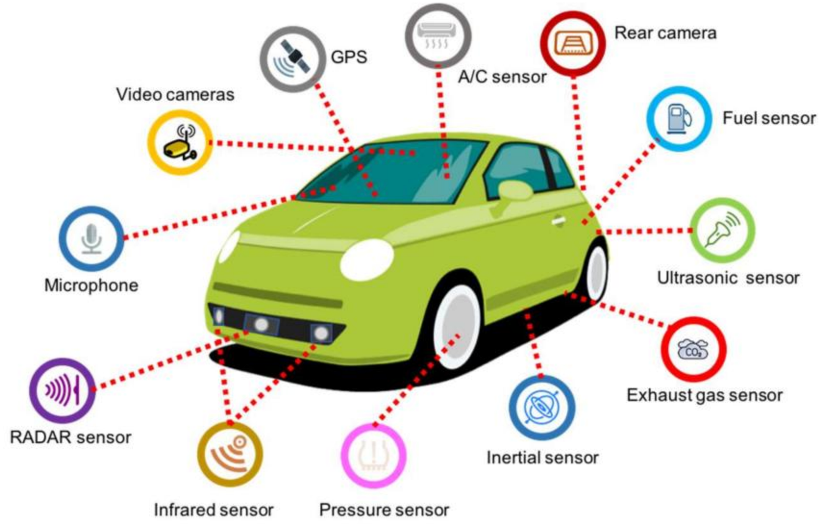
https://www.youtube.com/watch?v=5FDDDqz8MAM

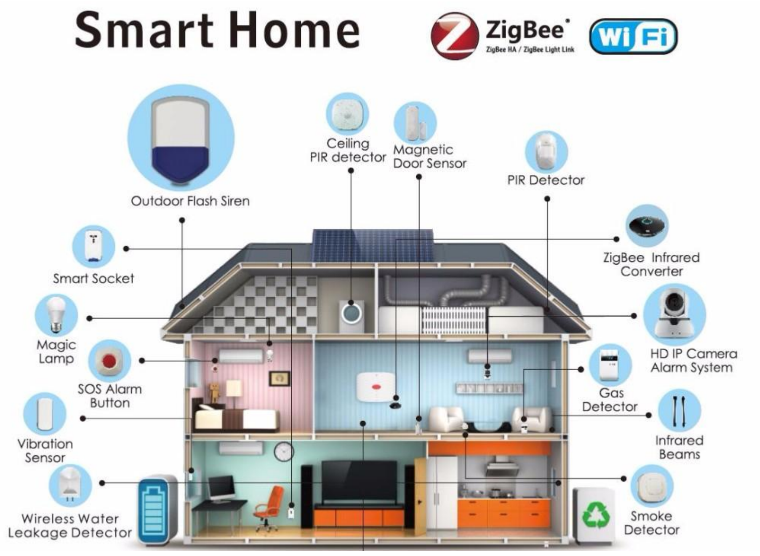
https://www.youtube.com/watch?v=K1WocU1iC6o

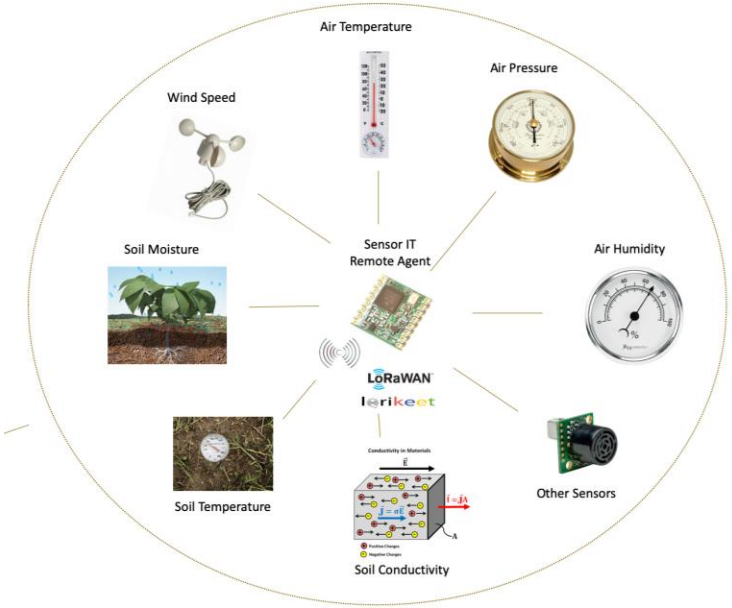
https://www.youtube.com/watch?v=Ile3DMzaRSA

https://www.youtube.com/watch?v=2Y3j4co_lBc

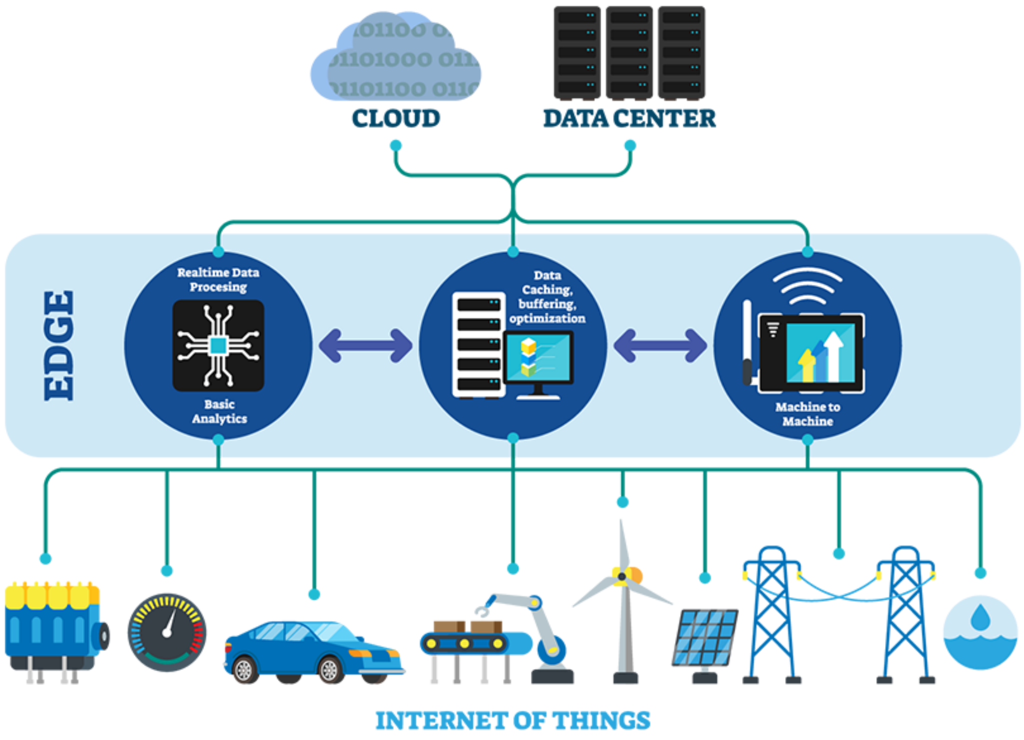
Internet of things

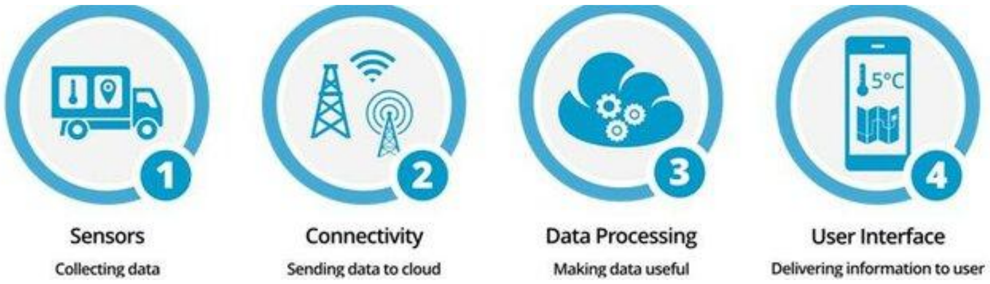


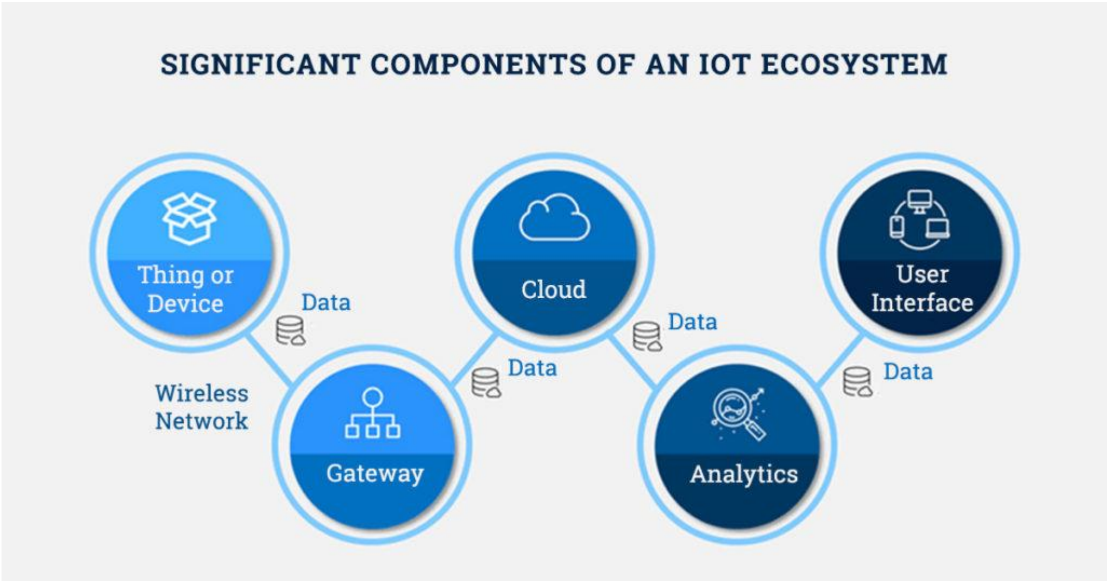




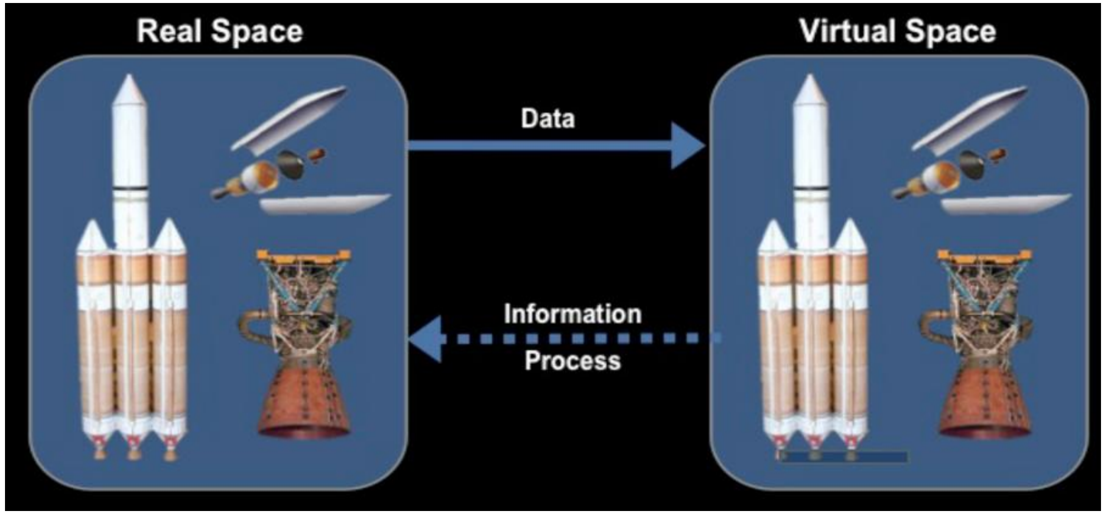
Internet of things

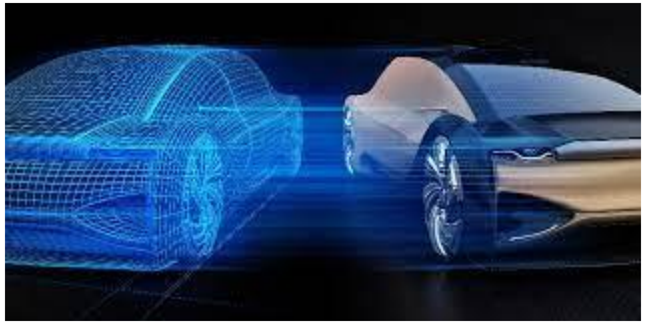






Digital twin/AR/VR



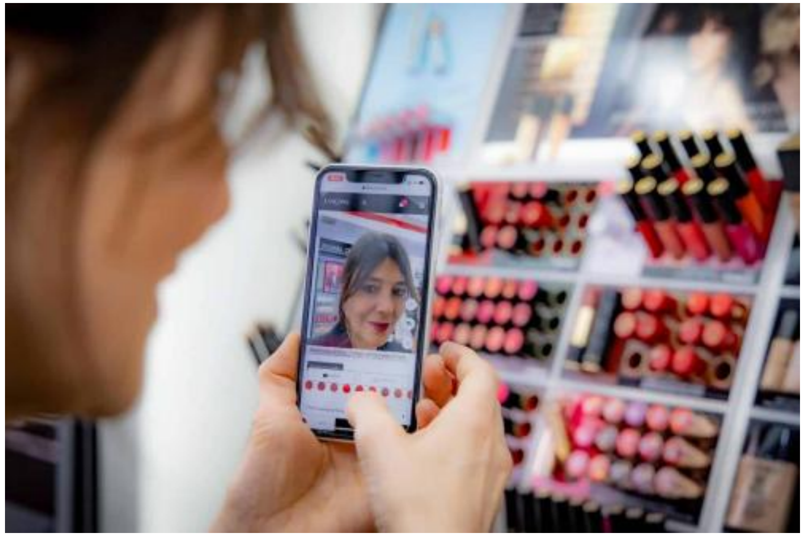


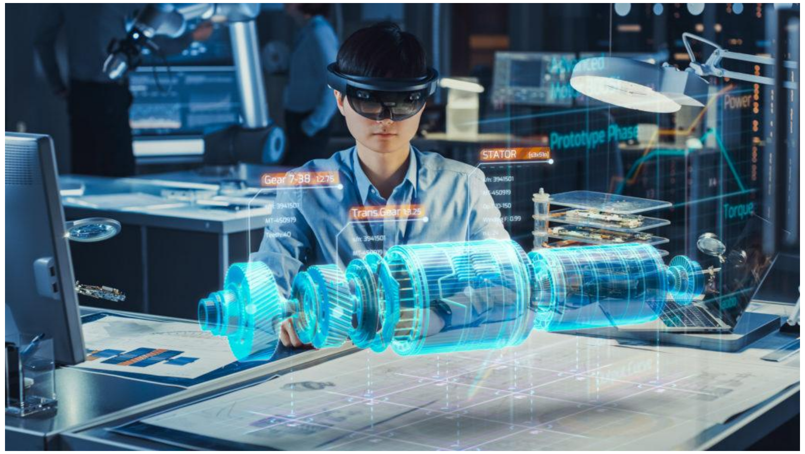
[Digital Twin of a Wind Turbine Power Train: TECNALIA]

28

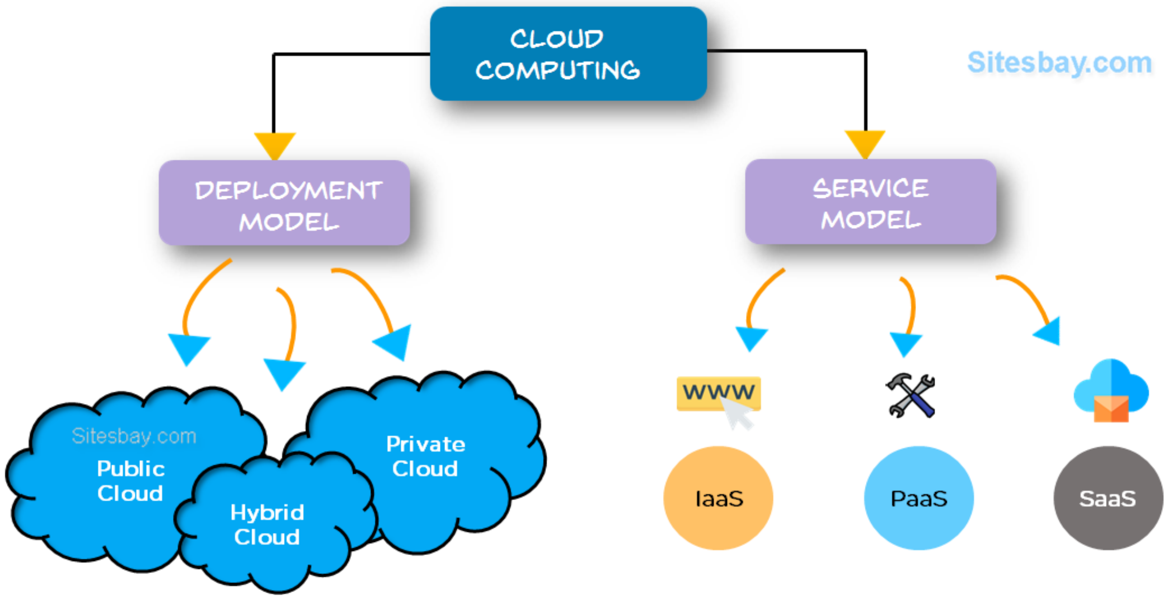
Digital twin/AR/VR

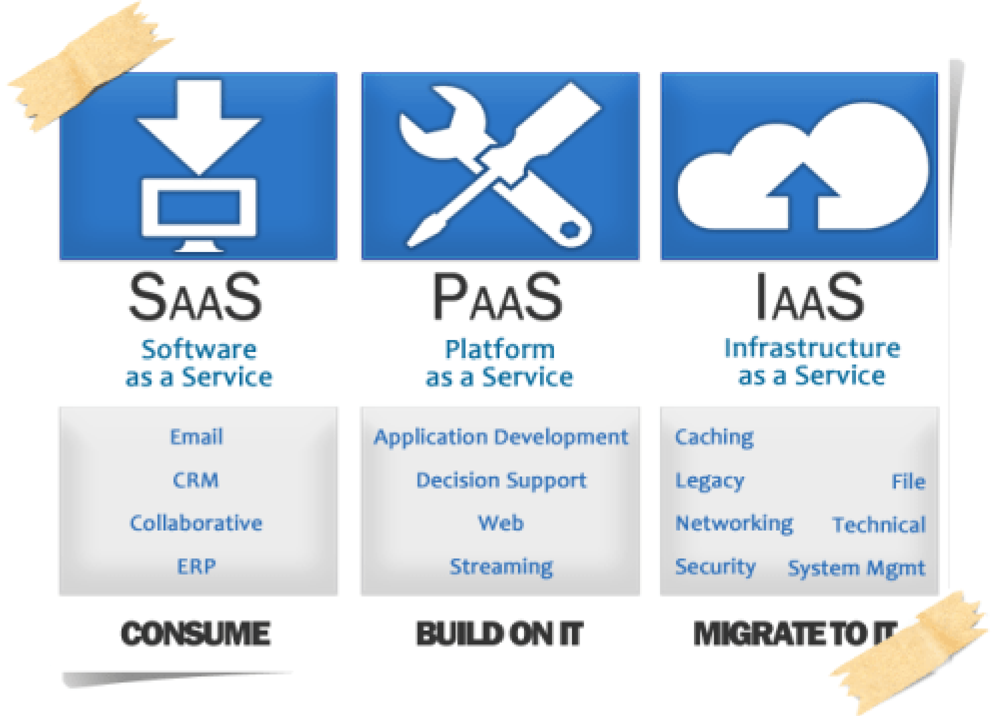
[Argos]



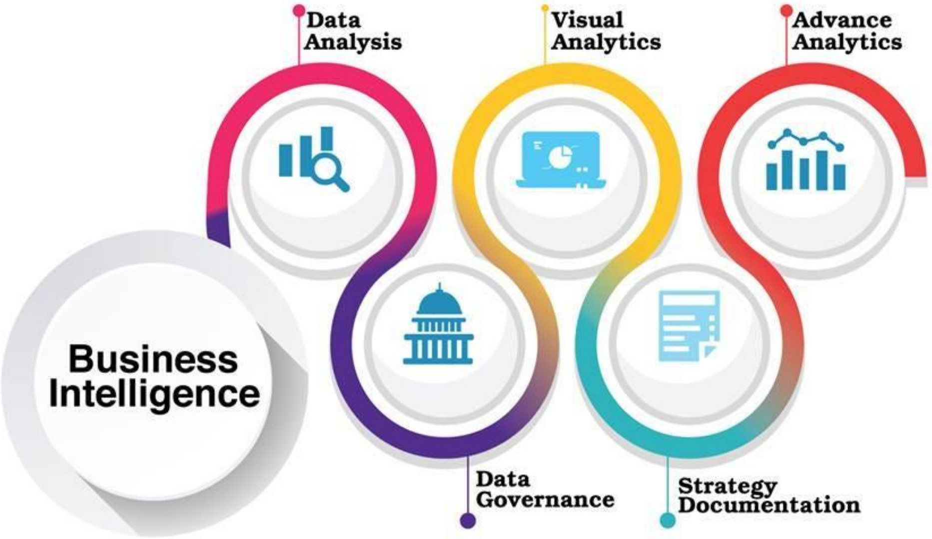


Cloud Computing





AI/Analytics



1970s

1997

2011

2016

2023

Evolution of Digital Technologies

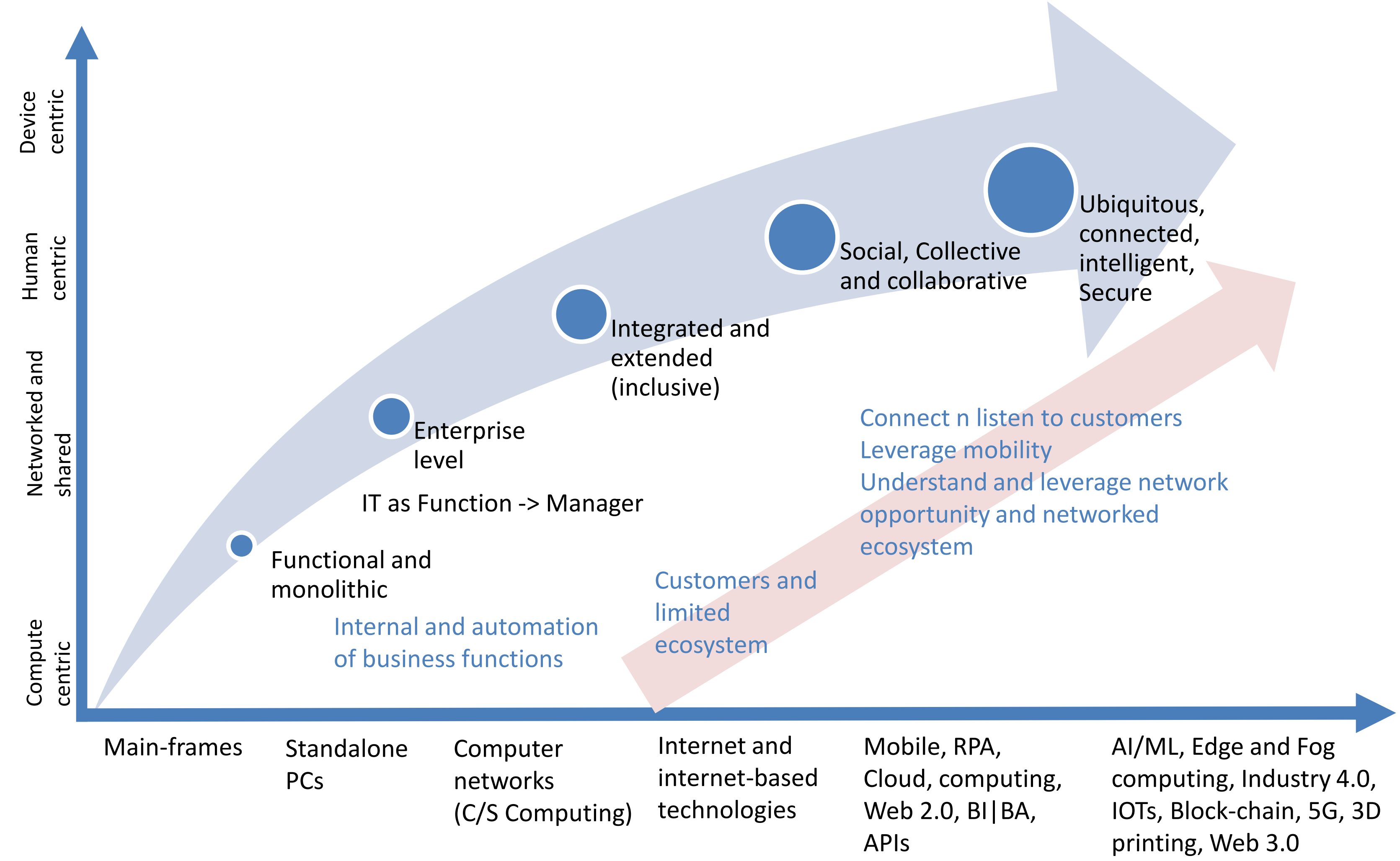
A Digitization
Records, transactions and basic MIS

B Automation
Workflows, ERP, portals and mobile apps

C Connectivity
Cloud, APIs, IoT and ecosystem integration

D Intelligence
Analytics, ML, GenAI and agents embedded in work

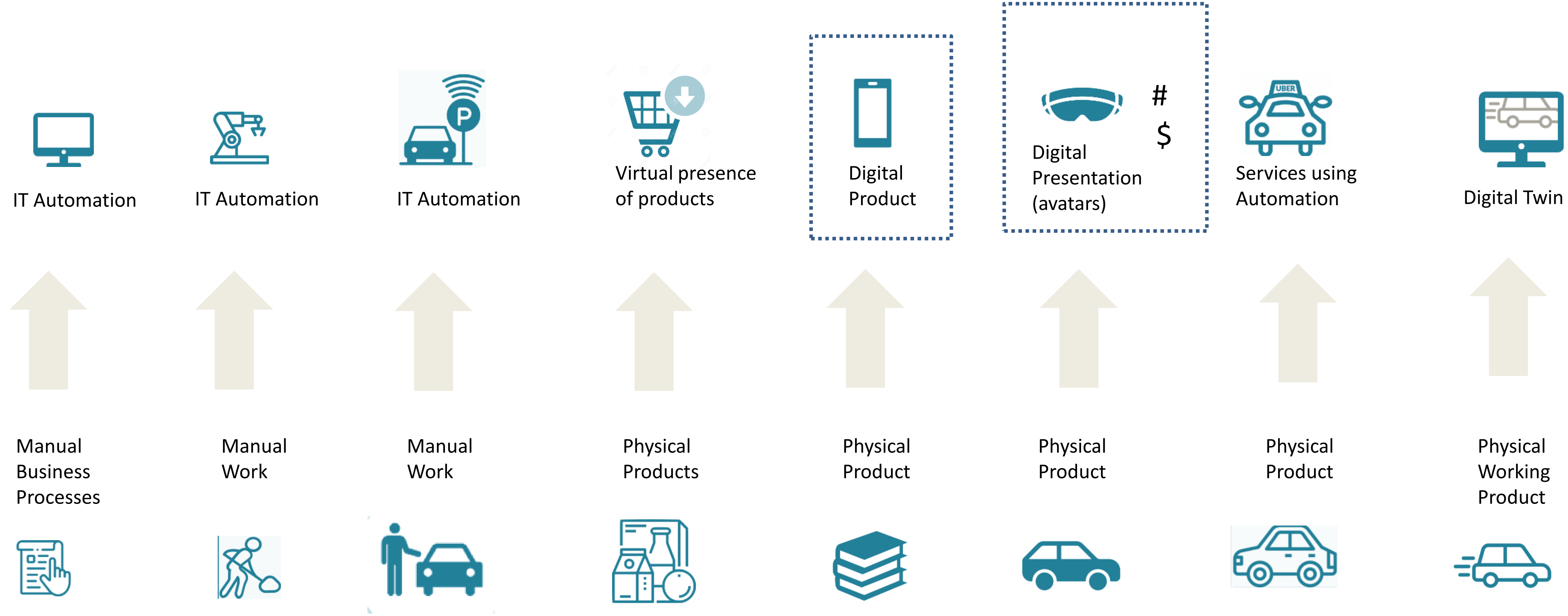
E Autonomy
Human-supervised adaptive processes and closed loops



Digital first and new normal
Be ready to be disrupted
Be resilient
Explore new business models
Rapid experimentation
Build digital capabilities

Automation + Robotics
Rise of Devices (- human intervention)
Virtual digital worlds
Physical + digital, Physical->Digital
Networked world and core competency
Data is everywhere
Infinite utility computing and storage: Cloud
Need for a platform
Intelligence everywhere

Evolution of Digital Technologies



Digital objects: Searchable | Indexed | Hyper Links | Editable | Omni-Channel | Responsive | Embed Objects | Sharable | Navigation| Downloadable | Personalizable | Convenience in usage | Customizable | Augmentable | Summarized | Contextual | Compatibility | Ad ready…

Trial | AR | VR | Customizable |…

$ Collaborate | Share | Innovate with Partners

AI can be integrated at deeper level: highly personalized and contextual

Evolution of Digital Technologies

Smart Asset Management

Decentralised 3D Manufacturing

Personal Printing

Virtual Meeting

Microservices

Personal Movie Viewing

Individual Movie Creation

Automated Content Generation

Manual Asset management

Centralised Manufacturing

Centralised Printing

Centralised Physical Meeting

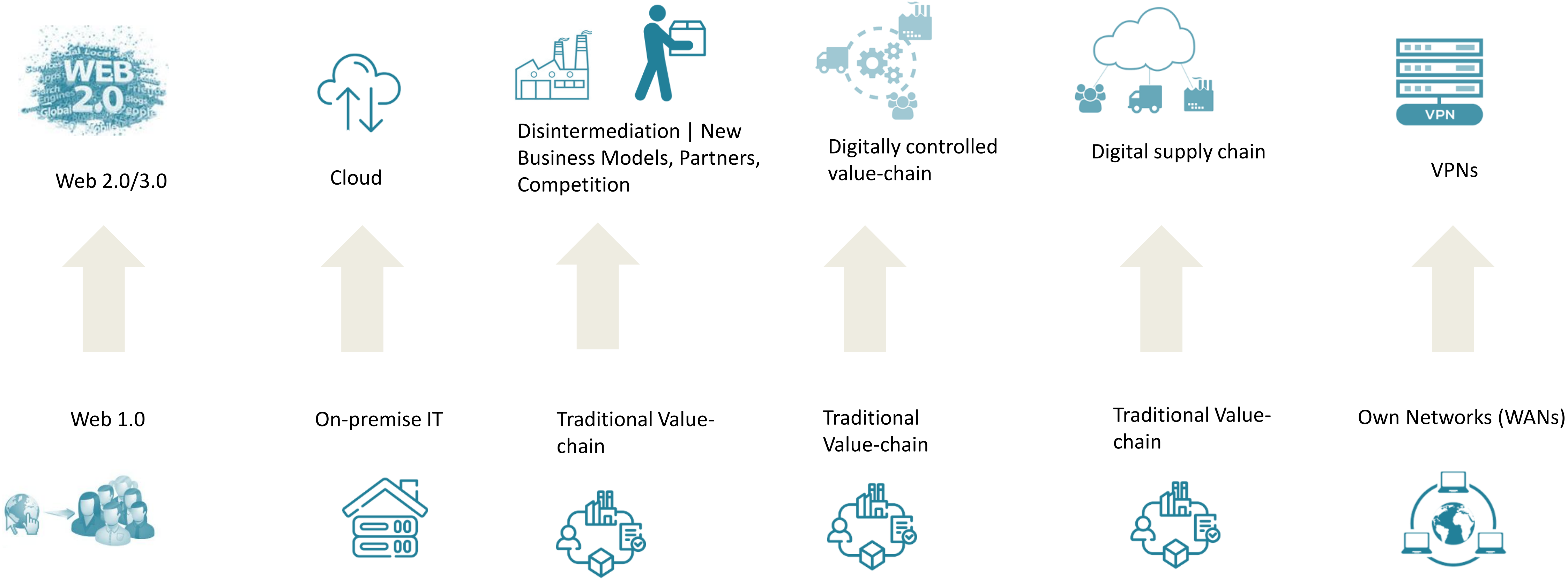
Monolithic Code

Theatre Cinema

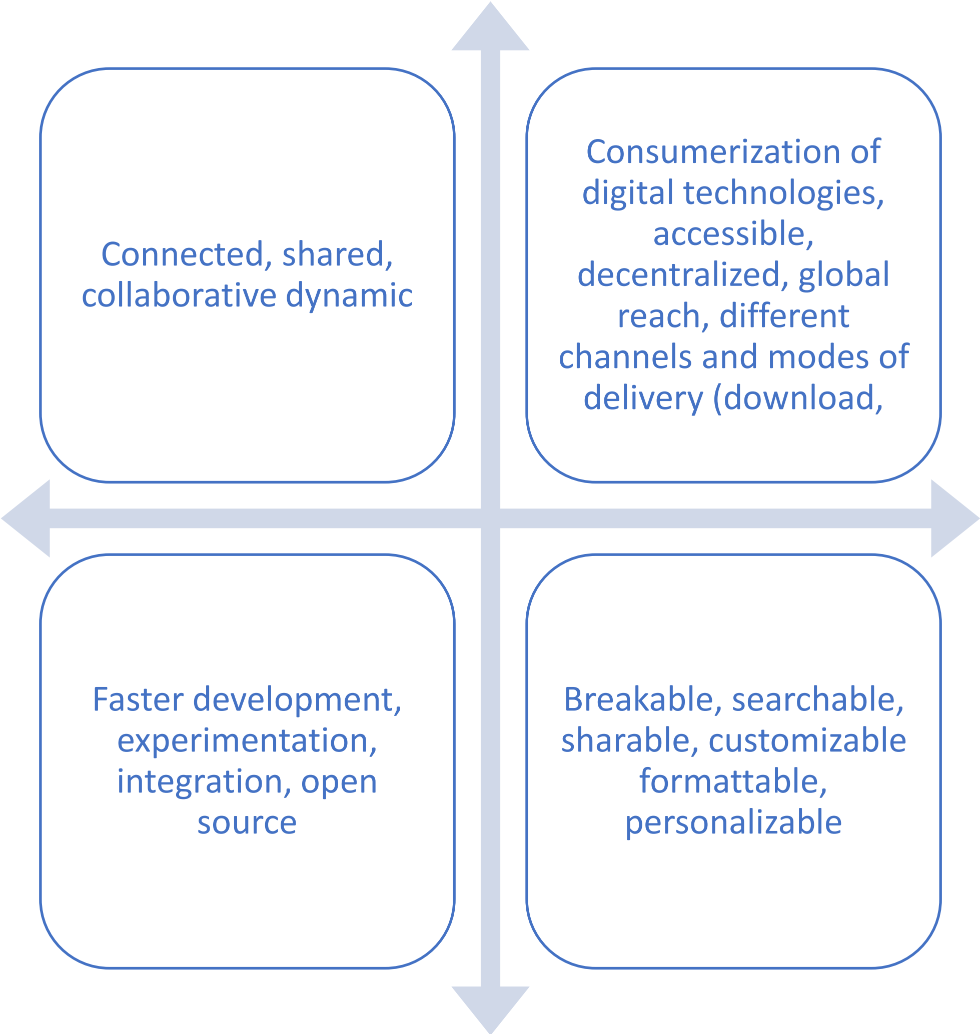
Movie Production House

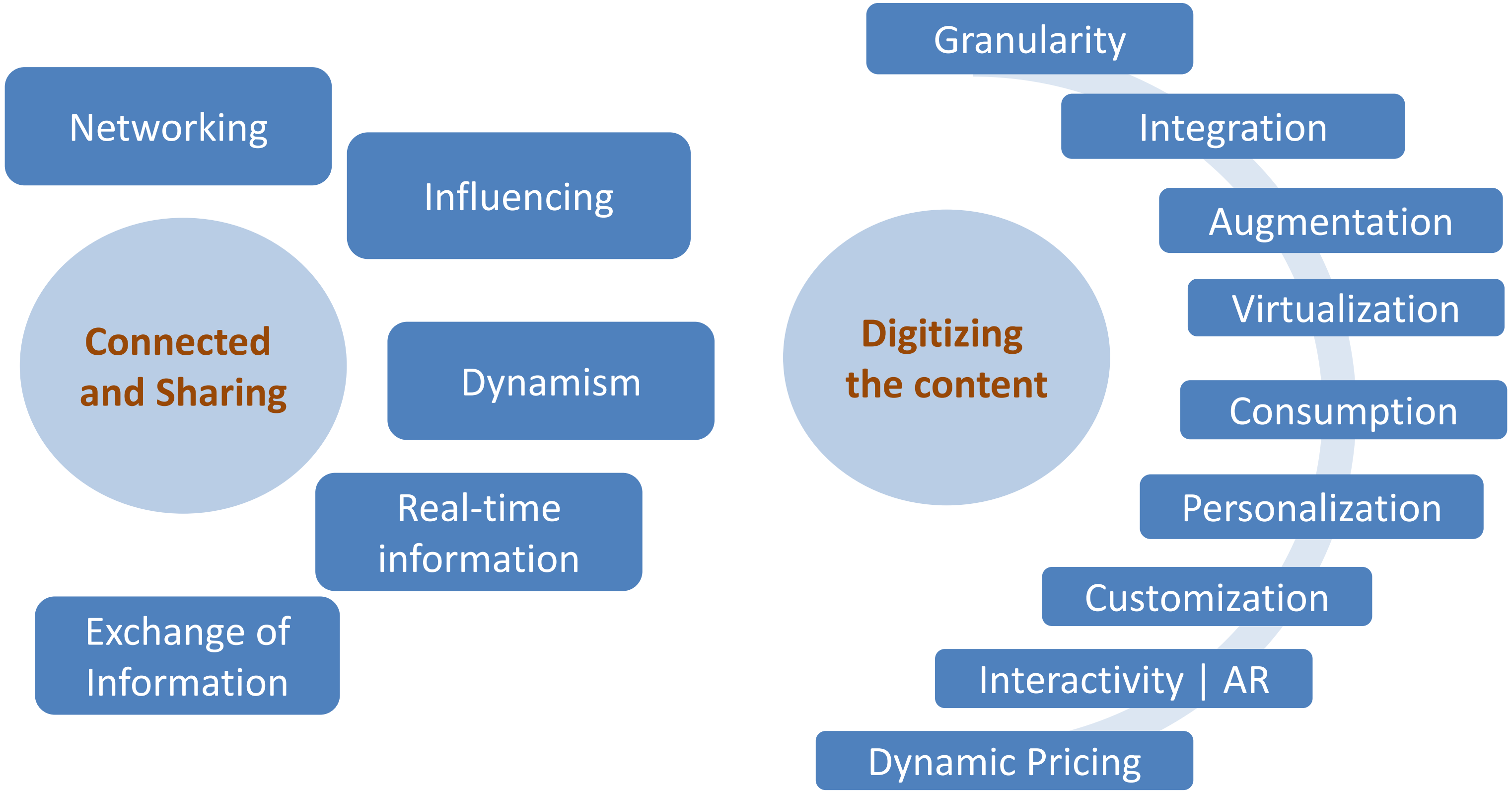
Manual content creation

Evolution of Digital Technologies



Evolution of Digital Technologies





AI can be integrated at deeper level: highly personalized and contextual on multiple dimensions

Why AI Needs Digital Foundations

AI is not a standalone technology project

AI fails when foundations are weak

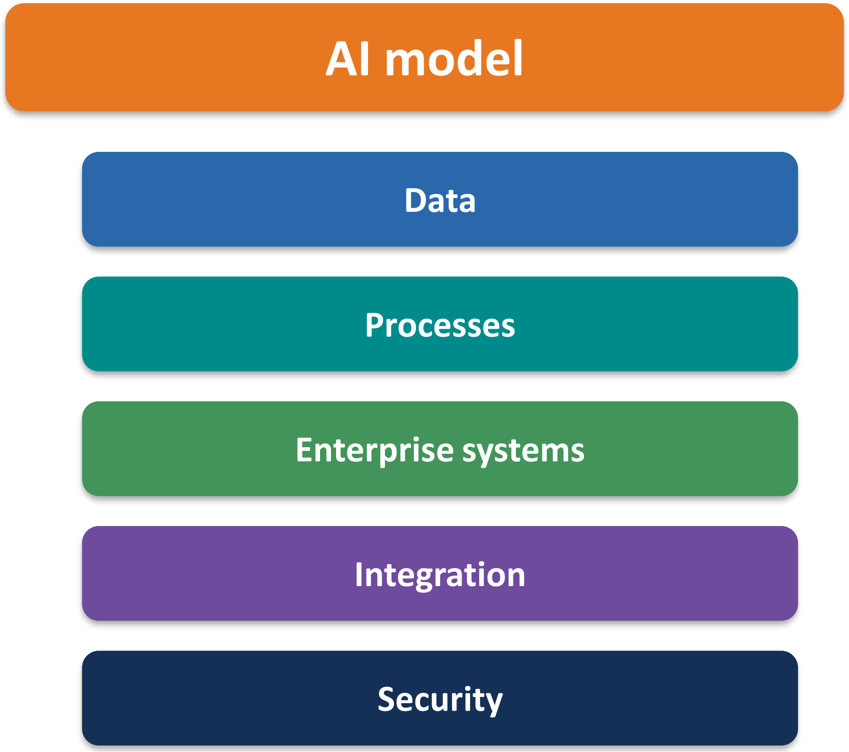
Data must be available, reliable and governed

Business processes must be clear enough to embed AI decisions

Systems must be integrated so AI can act on current information

Infrastructure must support scale, performance and resilience

Security and privacy controls must be built into the digital architecture



Business value requires the full stack

Digital Technology Primer for Leaders

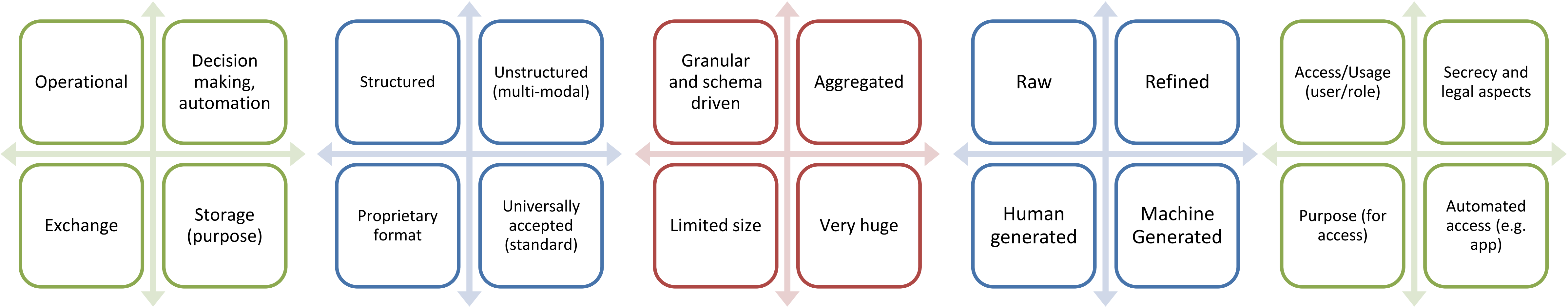
A shared vocabulary for AI-ready decision-making

User channels	Web, mobile, portals, chatbots
Application layer	Business logic, workflows, rules
Integration layer	APIs, events, messages, middleware
Data layer	RDBMS, NoSQL, files, lakes, warehouses
Infrastructure layer	Cloud, networks, compute, storage
Trust layer	Security, privacy, governance, resilience

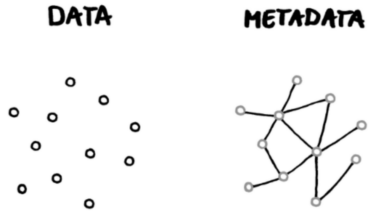
Leadership takeaway: ask how the layers connect, not only which AI tool to buy.

Evolution of Data Management

Dimensions of data



Metadata about data plays very important role. Just imagine you have all information about every type of data piece you have in one place and governed! Try to make data as much as granular, structured, understandable, meaningful and relevant to context*!



*We will talk more of bout it in AI data ready topic.

Data plays very crucial role, its responsible use, access policies and governance are very important.

Evolution of Data Management

From record-keeping to analytics and AI

Files	RDBMS	Data warehouse	Big data	Data lake	Lakehouse
Data files (e.g. CSV, COBOL), spreadsheets,	Structured transactions and operational data	Repository of enterprise level historical data, Reporting, dashboards	Volume, variety and velocity at scale	Raw data across formats and sources	Flexible storage plus warehouse-like reliability

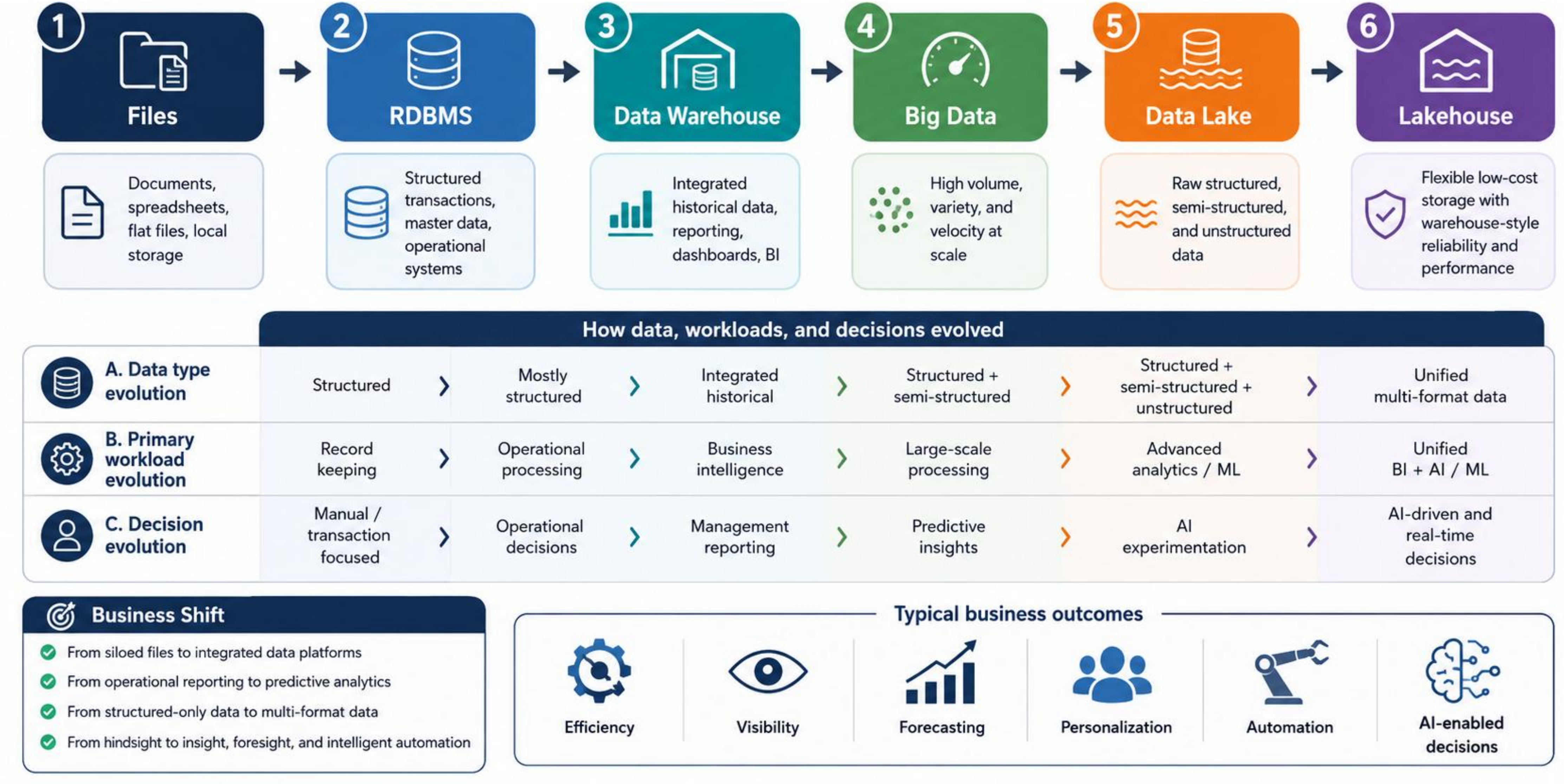
- Basic type of data (structured-unstructured)
- Operational data for real-time processing v/s decision making
- Storage architecture based on volume and data access
- Distributed v/s Enterprise view

Major factors behind evolution

The strategic shift: data is no longer only a by-product of operations; it is a reusable asset for automation, analytics and AI.

Evolution of Data Management

From record-keeping to analytics and AI



Common Data Formats

AI-ready organizations handle many kinds of data

CSV	XML	JSON
Simple tabular files Exports, spreadsheets, reports	Hierarchical structured data Legacy enterprise and govt systems	Lightweight API-friendly data Modern web and mobile apps

PDF / Docs	Images / Audio / Video	IoT / Logs
Unstructured business documents Contracts, invoices, policies	Rich media for modern AI Vision, speech and Gen AI	Continuous machine data Sensors, telemetry, monitoring

Structured, Semi-Structured and Unstructured Data

The data type influences storage, quality checks and AI use cases

Structured	Semi-structured	Unstructured
Examples ERP tables Transactions Customer records	Examples Emails Logs Customer care	Examples PDFs Images Videos Voice
Typical AI use Reporting Dashboards ML models	Typical AI use Integration Events Application data	Typical AI use Gen AI Document intelligence Computer vision

Modern AI increasingly combines all three: for example, a loan decision may use forms, transaction tables, uploaded documents and external API data.

RDBMS and SQL

The language of much of enterprise data

Core idea

RDBMS stores data in tables with rows, columns and relationships

SQL queries, filters, joins, aggregates and updates business data

Good for transaction systems that need consistency and auditability

Common in ERP, CRM, finance, HR, inventory and banking systems

```
SELECT customer_id, SUM(amount)
FROM orders
WHERE order_date >= '2026-01-01'
GROUP BY customer_id;
```

Business takeaway
Before AI, leaders must know which data is official, how it is structured, and whether it can be trusted.

NoSQL and Modern Data Stores

Not all useful data fits neatly into rows and columns

Document DB

Flexible JSON-like records
Product catalogs, profiles

Key-value

Very fast lookup
Sessions, cache, configuration

Graph DB

Relationships and networks
Fraud, recommendations, knowledge graphs

Time-series

Data ordered by time
IoT, telemetry, monitoring

Leadership lens: choose data technology based on the business problem, not technology fashion.

Web-Based Architecture

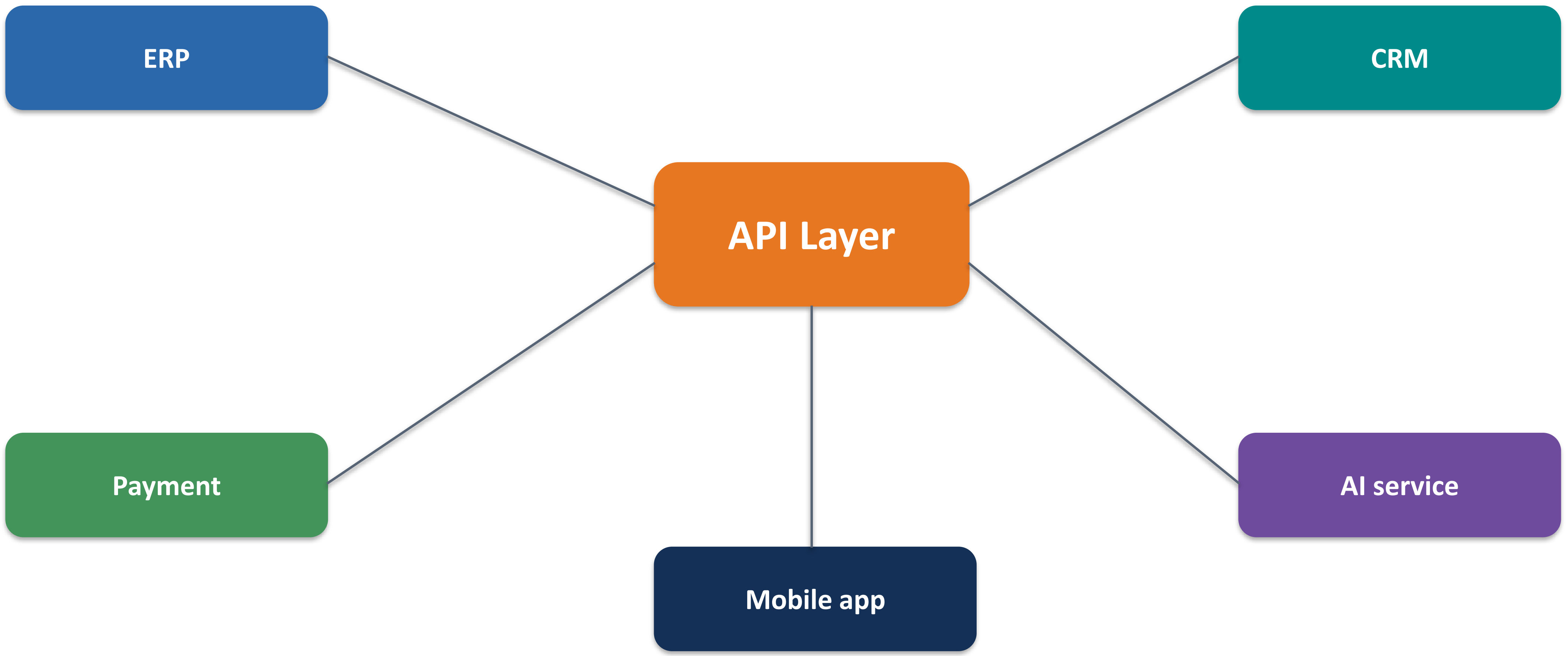
How a modern digital application usually works

User interface	Application server	API layer	Data & services
Web / Mobile / Portal	Business logic and workflows	Controlled access and integration	Databases, AI, external systems

AI can be embedded at different layers: in the user experience, in business logic, inside workflows, or as a service called through APIs.

APIs: The Connective Tissue of Digital Organizations

APIs allow systems, partners and AI services to communicate in controlled ways



Example: a loan approval AI may need APIs for customer profile, credit score, documents, policy rules, transactions and notifications.

From Systems of Record to Systems of Intelligence

AI creates value when it connects to existing enterprise systems

Systems of Record	Official business data ERP, CRM, HRMS, Core Banking
Systems of Engagement	User interaction channels Web, mobile, portals, chatbots
Systems of Automation	Workflow execution BPM, RPA, rule engines
Systems of Intelligence	Learn, predict, recommend ML, Gen AI, analytics, agents

Leadership question: where will AI sit in your enterprise architecture and which decisions will it influence?

AI, IoT, Cloud and Data in Digital Transformation

Four technologies often work together rather than separately

IoT

Captures signals from machines, products, spaces and operations

Cloud

Provides scalable compute, storage and managed digital services

Data

Connects historical, real-time, structured and unstructured information

AI

Predicts, recommends, generates, detects and automates decisions

Illustration: predictive maintenance = IoT sensor data + cloud storage/processing + analytics + AI prediction + maintenance workflow.

[Starbucks Vision Keynote Demo // Microsoft Build 2019]

Why AI Must Connect to Business Processes

A prediction is not value until it changes action

Sense	→	Understand	→	Decide	→	Act	→	Learn
Capture data or event		Analyze context		Recommend or approve		Trigger workflow		Measure outcome

Examples: fraud alert routed to investigator, inventory forecast converted into purchase order, churn prediction converted into retention action

Digital Readiness Checklist for AI

Questions executives should ask before scaling AI

- ✓ Do we know where critical data resides?
- ✓ Is the data clean, accessible and governed?
- ✓ Are systems integrated or fragmented?
- ✓ Do we have APIs for key business capabilities?
- ✓ Can workflows be automated end-to-end?
- ✓ Is infrastructure scalable and resilient?
- ✓ Are privacy, security and audit controls in place?
- ✓ Can AI outputs be embedded into decisions?

Discussion Exercise

Map one AI opportunity to its required digital foundations

Choose one business problem from your organization and map the foundations required for AI adoption.

Business decision

What decision or action will improve?

Data needed

What data sources and formats are required?

Systems involved

Which enterprise systems must connect?

Integration

Which APIs/events/workflows are needed?

Controls

What privacy, security and audit controls matter?

Key Takeaways and Bridge to Session 2

Digital foundations determine whether AI can scale

Session 1 takeaways

AI must be connected to data, systems, workflows and controls

Data management has evolved from files to databases, lakes and lakehouses

SQL, NoSQL, APIs and web architecture are foundational vocabulary for leaders

AI-ready organizations treat integration, automation and security as strategic capabilities

Next session: Integration, automation and security

How APIs, microservices and interoperability enable enterprise AI

How automation turns AI insight into workflow action

How infrastructure, privacy, security and resilience shape responsible AI adoption