AISWITCH 7-Layer Architecture Framework for Enterprise AI-automation Solutions

Who should read this: End-user Leaders- AI users/ strategists/ digital solution architects

Enterprise AI-automation leaders and end-users/ service providers/ business leaders/ solution architects who are planning to productionize and scale up AI-automation solutions.

Why a generic, pre-validated architecture framework for enterprise AI-automation solutions, is a must-have

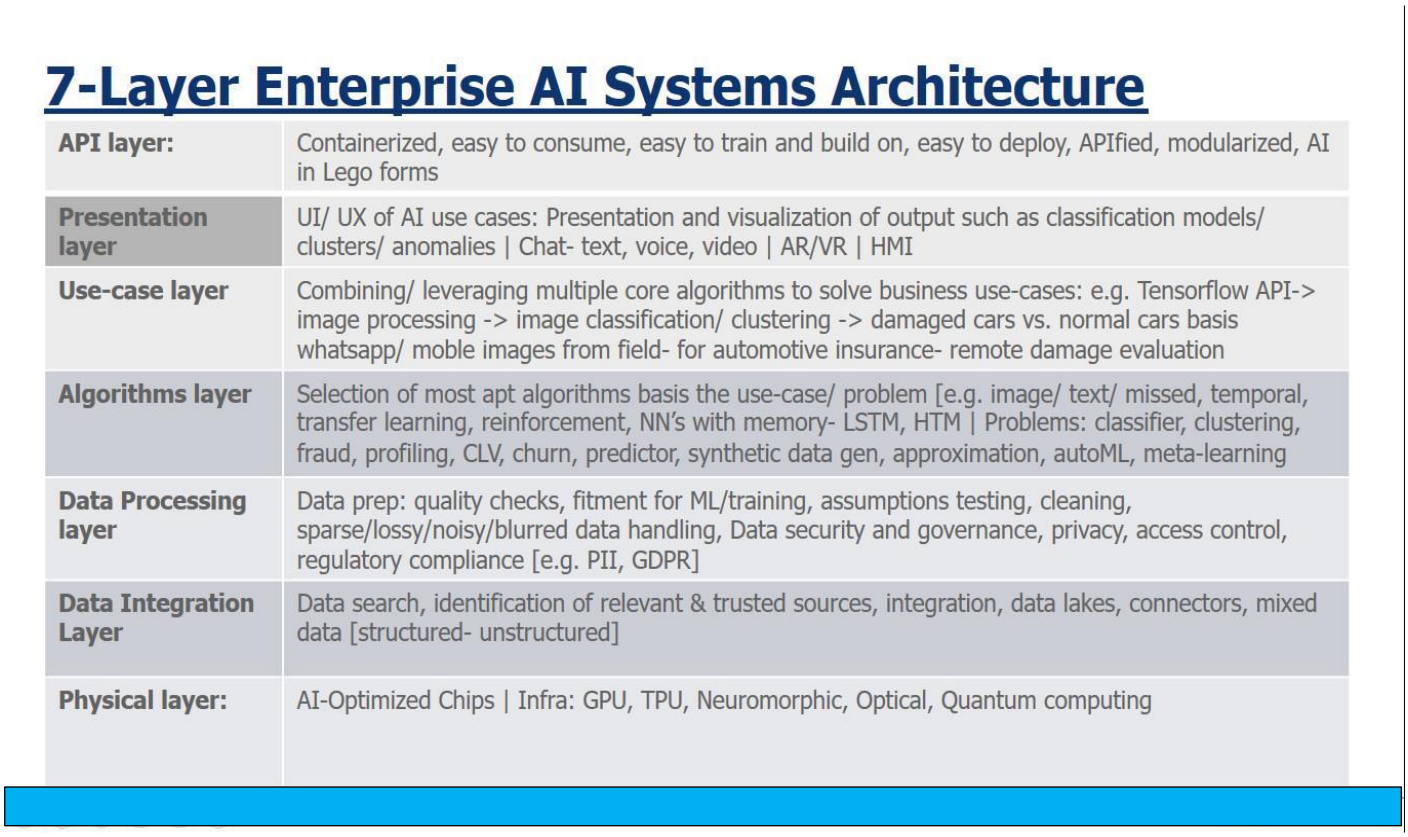
7-layer ISO-OSI architecture for computer networks is a tried and tested framework that we all tech practitioners are very familiar with. Drawing an equivalent of this generic architecture model to define and design enterprise AI-automation platforms & solutions, has several advantages as an approach, e.g.

- It offers a pre-validated and functional architecture framework for enterprise systems consisting of hardware, data management and systems and application software for different purposes, and interface and application management. As all network systems have these elements of data, logic and interface, enterprise AI-automation systems also have these fundamental Lego block components of data, logic/ algorithms/ applications, and presentation/ visualization/ consumption layers. Therefore, the analogy is quite rational and obvious.

- The generic 7-layer ISO/ OSI network architecture has a high degree of familiarity i.e. most engineers and software professionals who have had foundational training in these and related subjects, have a thorough understanding of these and subsequent specifics. This familiarity makes this framework easy-to-relate-to, for technology professionals including enterprise AI-automation solution/ technical architects.

- The advantages of using a generic framework to design enterprise AI-automation systems in a holistic, inclusive yet simple manner, ensures that the architects and designers don't miss out on any important aspects of these solutions, while also providing a uniform and consistent approach, irrespective of the underlying tech/tool-specific stacks or IDEs.

What does a 7-layer enterprise AI-automation architecture framework look like, and how to build it?



7-Layer Enterprise AI Systems Architecture

API layer:	Containerized, easy to consume, easy to train and build on, easy to deploy, APIfied, modularized, AI in Lego forms
Presentation layer	UI/ UX of AI use cases: Presentation and visualization of output such as classification models/ clusters/ anomalies | Chat- text, voice, video | AR/VR | HMI
Use-case layer	Combining/ leveraging multiple core algorithms to solve business use-cases: e.g. Tensorflow API-> image processing -> image classification/ clustering -> damaged cars vs. normal cars basis whatsapp/ moble images from field- for automotive insurance- remote damage evaluation
Algorithms layer	Selection of most apt algorithms basis the use-case/ problem [e.g. image/ text/ missed, temporal, transfer learning, reinforcement, NN's with memory- LSTM, HTM | Problems: classifier, clustering, fraud, profiling, CLV, churn, predictor, synthetic data gen, approximation, autoML, meta-learning
Data Processing layer	Data prep: quality checks, fitment for ML/training, assumptions testing, cleaning, sparse/lossy/noisy/blurred data handling, Data security and governance, privacy, access control, regulatory compliance [e.g. PII, GDPR]
Data Integration Layer	Data search, identification of relevant & trusted sources, integration, data lakes, connectors, mixed data [structured- unstructured]
Physical layer:	AI-Optimized Chips | Infra: GPU, TPU, Neuromorphic, Optical, Quantum computing

We all have studied the 7-layer ISO-OSI model of computer networks. Interestingly, the 7 layers, in way of the actions that they are supposed to perform, map quite nicely to the multiple layers of any enterprise AI system.

Here is how:

- **API layer:** [Equivalent of application layer in networks]- This offers either core platform/ technical algorithmic capabilities or applied/ use-case based capabilities in a Lego-blocks like components/ micro-services/ connectors form, making various AI services:
    - Containerized,
    - easy to consume,
    - easy to train and build on,
    - easy to deploy, modularized.
- **Presentation layer** [Similar to presentation layer in networks]: This layer is responsible for providing the interaction interfaces between man and machine. It can be omni-channel, mixed content, with near real-time dynamic interface requirements. That way, for the UI/ UX of AI use cases, presentation/ visualization needs to be standardized:

- Presentation and visualization of output such as classification models/ clusters/ anomalies '
- Chat- text, voice, video | AR/VR | HMI

- **Use-case & governance layer** [in place of session layer]: This layer is NOT equivalent in function to the session layer in Network. However, this is the most important layer through which core algorithms and AI logical layers are consumed by various business processes and functions. This layer involves:

    - Combining/ leveraging multiple core algorithms to solve business use-cases: e.g. Tensorflow API-> image processing -> image classification/ clustering -> damaged cars vs. normal cars basis whatsapp / mobile images from field- for automotive insurance- remote damage evaluation
    - Governance functions take care of data security, data & algorithmic bias reduction techniques, explainability, interpretability, consumption of various algorithms/IP's, compliance to regulatory frameworks etc.

- **Algorithms layer:** This is the most technically relevant layer which determines the process flow, logic and models that will be used to build the target use-cases. This will involve:

    - Selection of most apt algorithms basis the use-case/ problem [e.g. image/ text/ mixed, temporal, transfer learning, reinforcement, NN's with memory- LSTM, HTM etc.]
    - for problems such as identification of classifier models, clustering, fraud, profiling, CLV, churn, predictor, synthetic data gen, approximation, AutoML, meta-learning

- **Data Processing layer:** Data is the most critical input to AI-ML algorithms. Hence this is a critical layer too. If this is not done well, all AI decision models will be like GIGO [Garbage-In-Garbage-Out]. Data preparation includes:

    - complex statistical and mathematical processing on the aggregated data [structured/ unstructured/ text/ image/ mixed/ IoT sensor data etc.] that's being massaged for training, testing and validation.
    - This will involve quality checks, fitment for ML/training, assumptions testing, sparse/lossy/noisy/blurred data handling.
    - This will also have to take care of data security and governance, privacy, access control, regulatory compliance [e.g. PII, GDPR] etc.

- **Data Aggregation Layer:** This will be mostly achieved by integrating existing data lakes plus establishing data links to all other relevant data sources and systems, in batch or real-time, depending on the temporal nature of the input training data and the use-cases. This will include processes for data search, identification of relevant & trusted sources; integration of existing data lakes including connectors to different data source systems; mixed data [structured-unstructured] extraction mechanisms; integrating data from edge and IoT devices/ sensors; formatting and preparing for quality checks and clean-up operations at next level.

Last but not the least is the **Physical layer:** This is the base machine infrastructure that's designed, built and optimized for running AI workloads, deploying massively parallel compute with big data storage integrations. This will include AI-Optimized Chips such as NVidia GPU's-Tesla, Volta, Ampere; Google TPU's; data-secured trusted / confidential AI infrastructure e.g. on Intel SGX with separate ‘memory enclaves’ for encrypted high-security data storage; Neuromorphic chips e.g. IBM TrueNorth and Intel’s Loihi processor containing 128 neuromorphic cores, 3 Lakefield (Intel Quark) CPU cores, and an off-chip communication network; and Optical Quantum computing chips.