Original Principles (2003)

Adhere to the Corporate Data Model

The corporate Data Model provides a high level insight into all data that is used in processes and applications.

Added Value:

- Information is stored centrally, simplifying access to it.
- Preliminary investigations can be performed faster, because insight is provided in the domains that are impacted by the project, and how they are related to other domains and applications.
- Project start-up is accelerated, and system exploration and management is more efficient.
- Reuse of interfaces and functionality is enabled.
- Lower costs and lower time to market.

Consequences:

- The coherence in information management is a responsibility of the ICT department. It defines the corporate data model in close consultation with the data owners.
- Projects take the corporate data model into account during their initial stages. Initial costs may be higher, but they can be recovered during implementation and management.

Application:
Every project shows which data from the corporate data model is used, recognizing the coherence of data. New data are added to the corporate data model.

New Principle (2007)

Adhere to the Canonical Data Model

Use an indirect translation when exchanging data; from the data format of the source system to the canonical data model, and from the canonical data model to the data format of the target system.

Added Value

- Dependencies between integrated applications reduces because the number of different translations decreases **and** changes only have impact on the system that is to be changed and the middle ware, not on the connected systems
- Better oversight and insight in domain overarching data
- Lowering the integration costs of (parts of) systems.
- The number of dependencies decreases. Changes will be simpler.

Consequences

- Of all shared data (e.g., flight, relation, service, cargo, visit) the definition in the Caonical datamodel is known on (shared) attribute level
- The SOA Competence center maintains this model, including version control and life cycle management
- Interfacing between systems always uses an indirect translation, unless there are exception criteria which are determined in advance
- Introduction of an extra infrastructure (translation) layer.
- New interfaces are translated indirectly. Existing interfaces are change on a natural moment from a direct translation to an indirect translation.
- In the indirect translation, the unique identification of delivering and receiving systems is created using reference tables.
- Deviations are allowed, provided that:
    - An indirect translation demonstrably does not meet the performance requirements.
    - AND there is 1 source and 1 target system. If an additional source or target system is added in the future, then the interface still has to be changed to the CDM.

Application

When interfacing, in connection with shared data

Arguments for changing the principle

From AP book:

- "Adhere to corporate data model" conflicts with principles "re-use before package selection" and "package selection before custom development"
- Databases were seldom shared between applications (*evidence*) and no common view on corporate data was achieved. (*evidence*)
- The actual definition of the corporate data model was never achieved (*evidence*) which undermines most of the other potential added value. (*evidence*)
- The emphasize should be on a clear and standard way of interfacing, and not so much on how an application stores its data internally. (*expert opinition*)
- The canonical data model focuses on standardizing the information that is exchanged between applications, instead of how information is stored.

From Schiphol principles document:

- The original principles has been used minimally in the previous four years in project evaluations. There is also no need to apply it now. (*evidence*)
- The principle is a principle in the area of information architecture.