

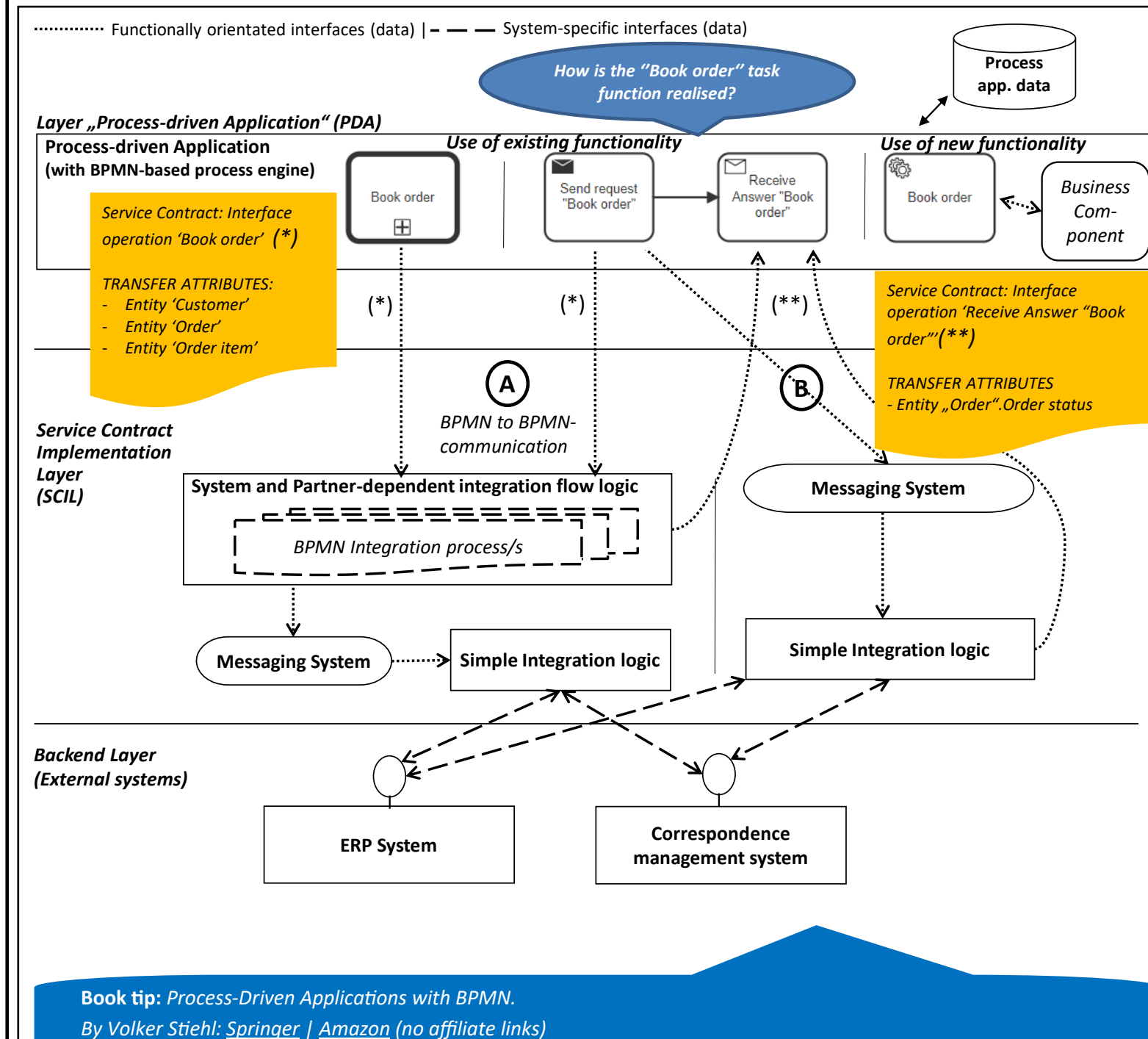
Cheat Sheet: Process-driven Approach by Volker Stiehl (Page 1)

Implement business process applications transparently, sustainably, efficiently, flexibly and with high quality

Definition:

The Process-driven Approach is a project management and implementation methodology based on BPMN process models for digital transformation projects for the realisation of any innovative business processes.

The Process-driven Architecture at a glance:



Process-Driven Application [PDA]

- A **Process-driven Application (PDA)** contains one or more business processes that generate competitive advantages for the company ('core processes'). Existing functionalities required in the business processes are requested from the system landscape via the *service contract* and thus reused. A PDA only works on its own data model: 'business data'.
- The **Service Contract** of the PDA defines the functional requirements of the business processes within the PDA for the system landscape. The service contract is *bidirectional*, in other words it is defined for service interfaces both from the PDA to the 'outside world' and from the 'outside world' to the PDA from the PDA's point of view and uses only the PDA's data model.
- The **Business processes** are described in *BPMN* in alignment with the standard and existing *best practices* by the business experts. These are then executed by a BPMN-based Process Engine.
- In **close cooperation between business experts and IT**, the TO BE is defined 'top down' without taking current organisational restrictions and systems into account, in other words starting from the business process and moving from the general to the specific using the top-down approach by *Bruce Silver*¹. All questions are derived from the business process (i.e. discussed with business experts): What data is relevant? - Which functionalities are required (existing vs. new functionality)? - Which rules need to be applied? - Which user interfaces need to be created?

Service Contract Implementation Layer [SCIL]

- Meaning of SCIL:** The functional requirements of the Process-driven Application for the existing functionalities in the system landscape ('outside world'), which are provided in the form of the service contract, are implemented by the SCIL. One or more application systems/services may be involved in the realisation of an existing functionality. On the other hand, new functionality is part of the PDA.
- Integration variants:** With at least two system calls, which can also affect the same system, there is a comprehensive integration process logic, which is labelled '*System and partner-dependent integration flow logic*' in the figure. To improve transparency, it is also recommended to implement this in BPMN and execute it in the same BPMN-based process engine [A] that is already used for the PDA processes. The classic integration tasks, such as data transformation and data mapping of the input data format to the target format, as well as the technical connection of the target systems, are all carried out in the integration component labelled '*Simple integration logic*'. If there is only one system call, this is a simple integration logic that can be handled directly by the '*simple integration logic*' (= integration component) [B].

[1] Silver, Bruce (2011): BPMN Method & Style. With BPMN Implementer's Guide. 2nd Edition

Cheat Sheet: Process-driven Approach by Volker Stiehl (Page 2)

Implement business process applications transparently, sustainably, efficiently, flexibly and with high quality

The **four golden rules** of the Process-driven Approach:

1. *Separate business processes from integration processes.*
2. *Never call systems directly in BPMN, e.g. via connectors.*
3. *Define Requirements for the systems ('outside world') via a service contract from the perspective of the Process-driven Application*
4. *Do not extend any application systems/backend systems, but retain them as they are.*

PDA layer: Relevant information for the development of business processes (overview)

- **Develop business process(es) in BPMN** using the "top-down" approach according to Bruce Silver¹
- **User tasks:** Present relevant information using information sketches ("pencil sketches"), wireframes or mockups.
- **Tasks with new or existing functionality:** Specify new functionality using the use case description (—>Service Task); Request the realization of existing functionality via the service contract of the PDA (—> Send Task/Receive Task/Call Activity)
- **Decision tasks ("Decision Tasks"):** Specify decision requirements and implement decision logic using DMN means (e.g. decision tables).
- **Functional (conceptual) data model for the PDA:** e.g. domain class diagram (UML) or entity relationship diagram (incl. attribute list)
- **Glossary:** Define terms of the PDA application domain
- **Define non-functional requirements in a measurable way:** e.g. with the help of quality scenarios

Die einzelnen „Gewerke“ im Überblick:

- **Process-driven Application (business process(es))** —> Configure process definitions/process models for the BPMN-based process engine, implement business components on a component basis
- **Decision models (DMN)** —> Implement DMN decision models using a decision management system
- **Frontend: User interfaces (UI)** —> Native app, web apps (e.g. based on React JS, Angular JS, ...)
- **Integration of existing functionality** —> messaging systems for asynchronous communication (e.g. Apache Kafka, RabbitMQ, Apache MQ Active Artemis, Azure Service Bus); integration software or own integration components using integration frameworks (such as in the Java environment: Apache Camel), configure/implement BPMN-based integration processes if necessary
- **Implement business components or microservice-based new business logic** in a programming language of your choice

Challenges of companies in the 21st century/Digital Transformation and how the process-driven approach can help

Challenges

The Process-driven Approach helps by:

Cost reduction in application development through efficient processes, maximum parallelism and reduction of technical debt	<ul style="list-style-type: none">* Vendor-neutral and industry-neutral approach based on standards such as BPMN, DMN & Co (independence).* First of all, focus on business processes or individual scenarios of a business process and the requirements to be derived, then enable the parallel development of the various "units" for the business process or scenario of a business process.* Reference architecture as a blueprint —> Focus: separation of business process from system integration details (basis for reducing technical debt).* Agile approach: Close cooperation between business and IT (fostering acceptance by business experts and efficiency in development).
Enabling innovation: The introduction of new products/services through creativity requires the rapid adaptation of processes	<ul style="list-style-type: none">* Focus on the company's own business processes (core processes) and accelerate development through this holistic and vendor-neutral approach as well as the basic technologies used, such as a BPMN-based process engine for execution. The reference architecture also helps with the above-mentioned challenges.
Differentiation from the competition through innovation and added value for customers	
Flexible response to continuous change such as new customer expectations, new competition, new market conditions and new legal framework conditions	<ul style="list-style-type: none">* Thanks to a living process model using the BPMN standard as a common language, the business process can be extended by business experts.* Thanks to the reference architecture, business experts can make enhancements to their business processes and thus to the process-driven application independently of specific systems.* Agile approach: Close cooperation between business experts and IT (product orientation).
IT's ability to act (IT as a driver of innovation) —> This requires "top-down" thinking, based on business processes	<ul style="list-style-type: none">* The Process-driven Application service contract enables IT to modernize its system landscape independently of the business processes and use innovative technologies to fulfill business processes. It only has to fulfill the service contracts of the Process-driven Applications.