

Designing cases using aggregates

Summary

This document discusses the most important concept of Dynamic Case Management (DCM), namely the **case**. We will describe how a case can be modeled as an aggregate in Blueriq. Furthermore, all DCM concepts will be mapped onto a Blueriq element, or a combination thereof. Finally, a generic design of a case aggregate is discussed, containing the entities to model a DCM case correctly.

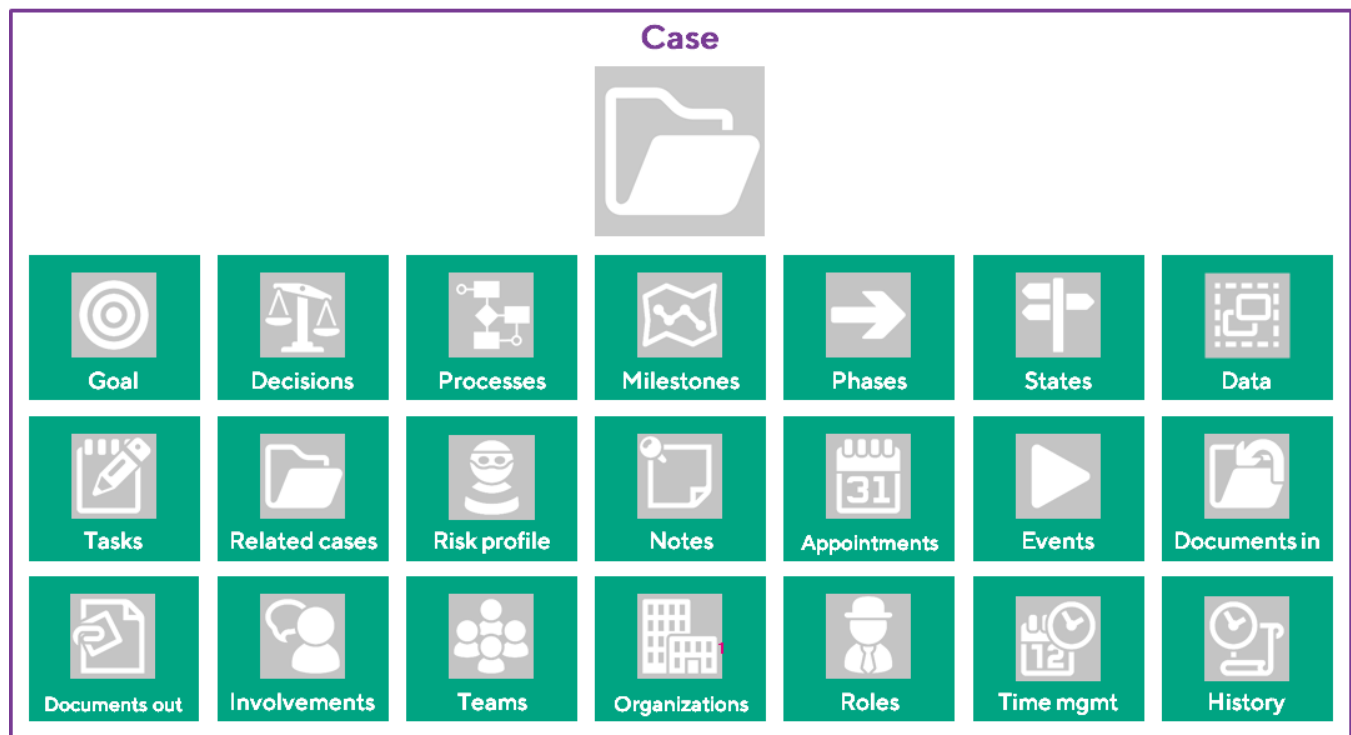
DCM concept of a case

Within DCM, the concept of a **case** (some use the more precise term case type, but we stick to case) is the most important concept.

In this design guide

- [Summary](#)
- [DCM concept of a case](#)
- [Blueriq concept of a case](#)
- [Comparison](#)
- [The generic case aggregate](#)
- [Entities in the case aggregate](#)
- [Relation between cases](#)
- [Characteristics of a case](#)
- [See also](#)

When asked what a case is, many people - even with similar background - tend to give many different answers, naming different concepts within the definition. Such concepts are decision, process, document, phase etc. This is due to the fact that a single definition of a case is very hard to give, but mostly because many different concepts play a role in DCM. All answers given by specialists have one thing in common: The named concepts are all part of the case, in other words they all contribute to a case. This is shown in the diagram below.



As shown above, in DCM a case is a concept that combines many other concepts. We do not aim to give one precise definition of a case in this document, but try to explain which concepts can play a role when designing a case. One misunderstanding can be corrected upfront: a case in terms of DCM is not just a process or an instance of a process¹. **A case is way more than just (an instance of) a process.**

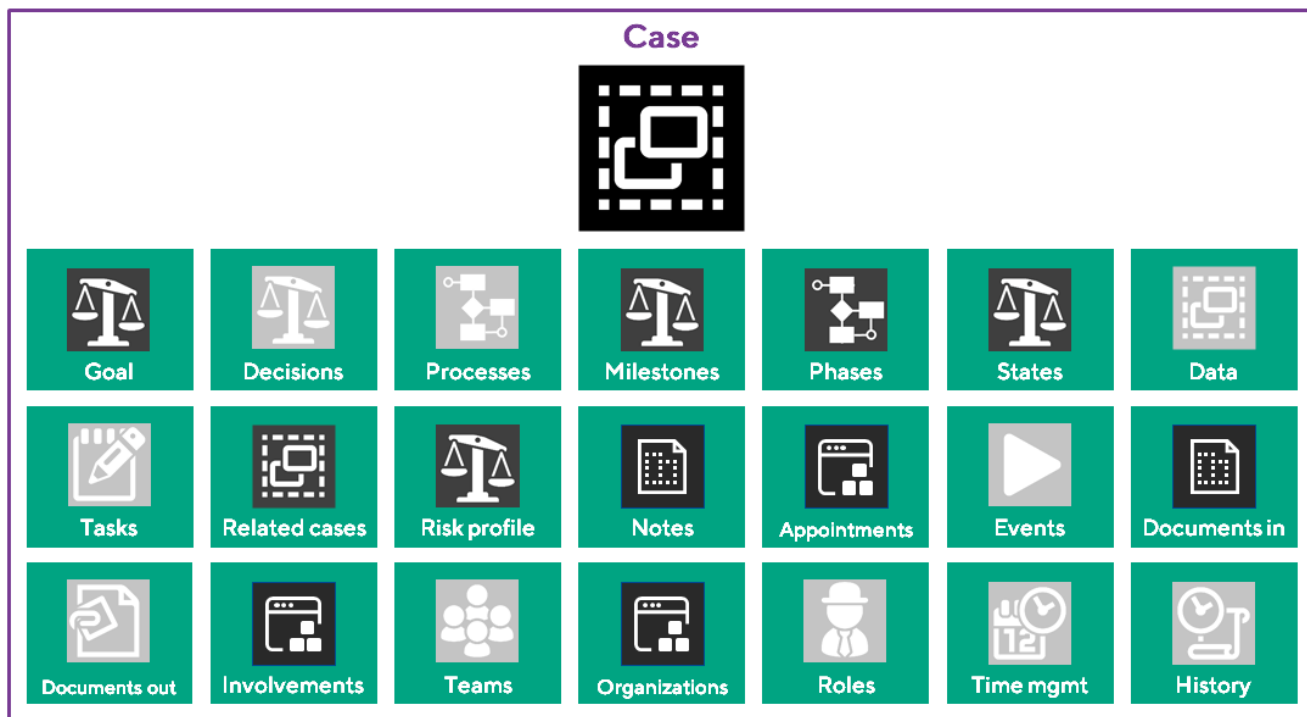
All concepts in the image above are briefly discussed below.

A case always has a goal, this is most likely singular. In order to reach that goal, many concepts play a role. For instance decisions are made, in one or more processes. Furthermore, a case goes through phases in time, where a phase change is achieved formally by means of reaching or crossing a formal moment called a milestone. A case can have different states, even more than one simultaneously. Data is persisted in a case and a case can use data from outside sources, such as a register. Within cases, tasks can be performed in processes, but also outside of the scope of processes. Such tasks are also called generic tasks and are part of the case definition as well. Cases may relate to other cases or can be hierarchical. In some domains, a risk profile is used to determine whether additional checks are needed. Notes and appointments are part of the case definition and are probably available throughout the life cycle of a case. Events can lead to actions that need to be performed in a case. Cases will most likely contain uploaded documents and will generate documents themselves as well. Cases are all about collaboration; people can be involved and teams, organizations and roles can be defined to execute parts of the processes within a case. Time management is almost always relevant, since law and legislation state maximum durations, for instance for deciding on an application. Finally, history is an important aspect of a case as well; it is necessary to see who did what and when.

All aforementioned concepts contribute to a case, but that does not mean that each case has to consist of all these concepts necessarily. However, most (or even all?) cases contain decisions, will have at least one process and will most likely store some data and/or produce a document.

Blueriq concept of a case

With Blueriq's proposition of Dynamic Case Management, many DCM concepts are available within the Blueriq software platform. However, not all concepts can be found in Blueriq as a standard out-of-the-box element. Shown below is the same diagram as above, containing a case and all its related concepts. The concepts in black deviate from the previous diagram; this means that the DCM concept itself is not available within Blueriq as a native Blueriq element, but one or more Blueriq elements can be used to model or compose the DCM concept effectively, completely and correctly.



In the diagram above, all **grey** concepts are DCM concepts that are native Blueriq elements. The decision is one of the most dominant ones, but also processes, data persistency by means of aggregates and output documents are used in many - if not all - Blueriq applications. Events, teams, roles, time management and history may not be as common as decisions, but they are used very frequently.









The **black** concepts in the diagram above indicate DCM concepts that are not available within the platform as a native Blueriq element, but can be modeled using out-of-the-box Blueriq functionality. For instance, the goal of a case, milestones and states; they can all be modeled using decisions, most likely decision tables or business rules. Notes and incoming documents can be modeled using specific containers. Appointments, involvements and organizations can be modeled with attributes.





































The most protruding blue icons are that of the case itself and related cases; aggregates are used to model cases. This means that not only standard persistency by means of registers is done using aggregates, but also the case itself is an aggregate! The aggregate is actually a really good fit to model a case. An aggregate combines entities and stores instances of these entities together. A case combines many elements as well and stores them together.

All cases are modeled as aggregates. However, not all aggregates are cases (an aggregate can also be used to model a register)

Comparison

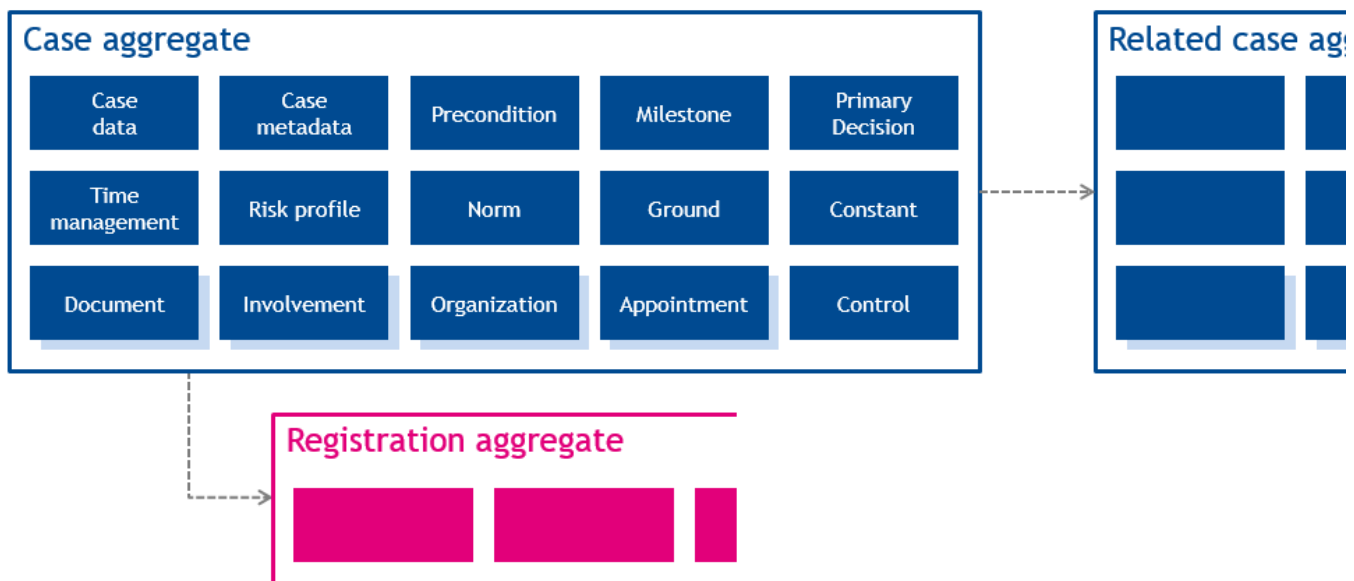
The table below compares all DCM concepts with Blueriq concepts and briefly explains how standard Blueriq elements can be used to model the DCM concept.

Concept in DCM	Concept in Blueriq	Remarks
 Case	 Aggregate	A case is modeled as an aggregate, meaning that for each case an aggregate is defined, containing all entities necessary to define a case. See also The generic case aggregate .
 Goal	 Decision	The goal of a case could be to grant a permit, register an applicant, pay a benefit, sell a product etc. A goal can easily be modeled by an attribute with a default value false and a decision table or business rule which sets the attribute to true when the goal is reached.
 Decision	 Decision	Decisions can be modeled using attributes, default value rules, reusable expressions, decision tables, business rules, external rules and data rules.
 Process	 Process	Processes can be modeled using the Blueriq element process.

	Milestone		Decision	Whether a milestone is reached or not is a decision, based on the context of the case. For more information on how to model milestones in DCM applications, see Designing dynamic processes .
	Phase		Process	With phases, a case can be broken down in large concepts, using the lowest granularity. Most cases consist of five plus or minus two phases. All phases are modeled as sub processes and since processes can be modeled hierarchically within Blueriq, the main process itself, containing the phases, is modeled as a process as well. For more information on how to model phases in DCM applications, see Designing dynamic processes .
	State		Decision	A state is a characteristic of a case that tells the end user not so much about the progress of the execution of the case over time (phases are used for that), but more about the progress regarding the case's identity. For instance, a case can have the state In progress. This does not really tell us how far we are, merely to be patient. Another example of a state is Waiting for customer, when additional data or documents are needed. States can be plural on occasion, so a case could be in two states, for instance Part of spot check and Confidential.
	Data		Aggregate	For persistency of data, aggregates are used. See also Persistency Management guide .
	Task		Task	Tasks are native Blueriq elements and can be modeled as such.
	Related case		Aggregate	Related cases are modeled as aggregates, just like the case itself. The relation between the related case and the main case is modeled as an attribute within the related case, containing the Id of the main case. The type of relation (parent-child, sibling, association, etc) can also be modeled using attributes. These relation-attributes are most likely part of a Case data or Control entity, but this is the designer's choice.
	Risk profile		Decision	A risk profile or risk calculation can be performed with decisions and sub-decisions perfectly, using potentially many attributes of the case.
	Note		Dashboard Comment container	Notes are modeled with a dashboard comment container. By default, the comments are multivalued and date-time and user are stored alongside the comment. If so desired, notes can be modeled with attributes of a Note entity, most likely when it is necessary to store more information alongside the note or when the note is typed (for instance Comment, Internal, etc.)
	Appointment		Attribute(s)	To model an appointment within a case, the best fit is to use attributes of an Appointment or Control entity. For the integration with a calendar (for instance Google calendar or Outlook), tailored software has to be bought or developed, since this is not available in Blueriq out-of-the-box.
	Event		Incoming external message	Events are situations from outside the case that require actions to be taken within the case. For instance the death of the applicant could require the case to cancel, or the divorce of an applicant could result in recalculation of a fee. In Blueriq, events are modeled as in incoming external messages. For an example of an incoming external message, see Designing dynamic processes .
	Document input		File upload container	File management is an integral part of Blueriq. Containers are available to upload and download documents to memory or the file system. Documents that are input for the case and need to be uploaded within the case, use the file upload container for that.
	Document output		Document	Output document can be modeled using document elements.
	Involvement		Attribute(s)	Involving persons in a case is not part of standard Blueriq functionality, but can easily be modeled using one or more (multivalued) attributes, most likely as part of an Involvement entity, but this is up to the designer. To involve someone in a case, for instance the applicant or the case handler or investigator, means setting an attribute with the desired involvement.
	Team		Team	Team is a native Blueriq element.
	Organization		Attribute(s)	Organizations strongly resemble teams, but an organization element itself is not available in Blueriq by default. Therefore, organizations can be modeled using one or more (multivalued) attributes, most likely part of an Organization entity, but this is up to the designer. To route a case to an organization, means setting an attribute with the desired organization.
	Role		Role	Role is a native Blueriq element.
	Time management		Time management	Time management is available by default in Blueriq, amongst others by means of (ad-hoc) timer nodes and the due date of tasks.
	History		Time line container	History can be modeled in Blueriq by means of a time line container.

The generic case aggregate

Since all cases can be modeled as aggregates, one might wonder what such an aggregate would look like. Are they completely different for each case or is there a common ground among all cases? The answer is the latter; aggregates that resemble cases all look alike. Shown below is the outline of such a generic case aggregate.



When modeling a DCM solution with Blueiriq, the use of this generic design of a case aggregate is strongly advised, just as the use of the DCM foundation library ([DCM Foundation](#)). The generic case aggregate is based on designs used in several projects, proof of concepts and demos and has proven to be successful thus far. It is suitable in government, insurance and financial domain where law and legislation play a pivotal role.

Entities in the case aggregate

The table below briefly explains each entity within the generic case aggregate.

Entity	Explanation
Case data	All data that is relevant within the case can be part of this entity. For example, in a case for an application for a permit to extend a house, all attributes of the application itself are case data, as well as the case handlers and relevant dates.
Case metadata	All metadata of a case can be combined in this entity. Example attributes are creation date, archive date, type, Id etc.
Precondition	Activities in a dynamic case are not modeled as part of a flow with gateways and arrows, but as ad hoc tasks with preconditions that tell when the activities may be performed (see also Designing dynamic processes). For purposes of understandability, insight and oversight, it is best to combine all preconditions of a case and make them part of one entity.
Milestone	For each phase within the case, a milestone tells whether the phase is complete and the next phase may begin. This entity contains an attribute for each milestone.
Primary Decision	The primary decisions of a case can be combined in this entity. Examples are whether an application is complete or granted, or whether a spot check needs to be performed etc. The goal of a case will most likely be part of this entity as well.
Time management	Time management or time monitoring can be part of the case when there are legal and/or legislative terms which must be complied with, such as a maximum amount of time in which an application has to be handled. Attributes used in this entity could be the terms themselves, but also the actual time that was needed.
Risk profile	A risk profile entity can be used when a certain permit, benefit or insurance may not be granted unconditionally, but is subject to for instance an audit or a two-men-rule, should the application pose a risk. Risk calculation could then be done using attributes throughout the case, the (intermediate results and) final result of such calculations should be part of this entity.
Norm	Norms are guidelines that are administered by law. People have to abide to the law, so they have to abide to or comply with the norms that are set within the law. For example, in an application case for a permit to extend a house, one has to comply with the norm that states that the additional surface of the extension is not more than a certain percentage of the surface of the original house.
Ground	Grounds are usually intermediate results, decided or calculated because of specific articles of the law. Examples are the ground for tax on employment income, franchises or calculated income on which pension is based. But also the surface of the intended extension of a house (as mentioned in the previous row) could be regarded as a ground and can be used to test whether a norm is complied with or not.
Constant	When there are constants in the domain of the case, it could be the designer's choice to explicitly model them as attributes of a constant entity. An example of such a constant could be the percentage that is used as a maximum regarding building extensions (as mentioned in the two previous rows). For more information regarding constants, see the chapter Constants of the Decision Management guide .

Document	With file management, it is advised to use a multiton (non-singleton) entity to create an instance for each document that is part of the application or the case. Such an entity contains metadata attributes like name, size, type, creationdate, etc.
Involvement	Involvements are modeled by means of a multiton entity, possibly containing attributes such as Id, role and name and from when to when (phase or date) someone was involved in the case.
Organization	When organizations are part of the case, the routing behavior as available for teams must be mimicked for organizations using this multiton entity. Additional tailored software needs to be developed to integrate teams in Blueriq's out-of-the-box routing mechanism.
Appointment	When appointments are part of the case, this multiton entity can be used to create multiple appointment instances. Possible attributes concern who wants to make an appointment with whom, when, where and why. Additional tailored software needs to be developed or bought to integrate appointments with a calendar application such as Google calendar or Outlook in runtime.
Control	This entity contains steering attributes, such as ProcessId, CaseId, TaskId.

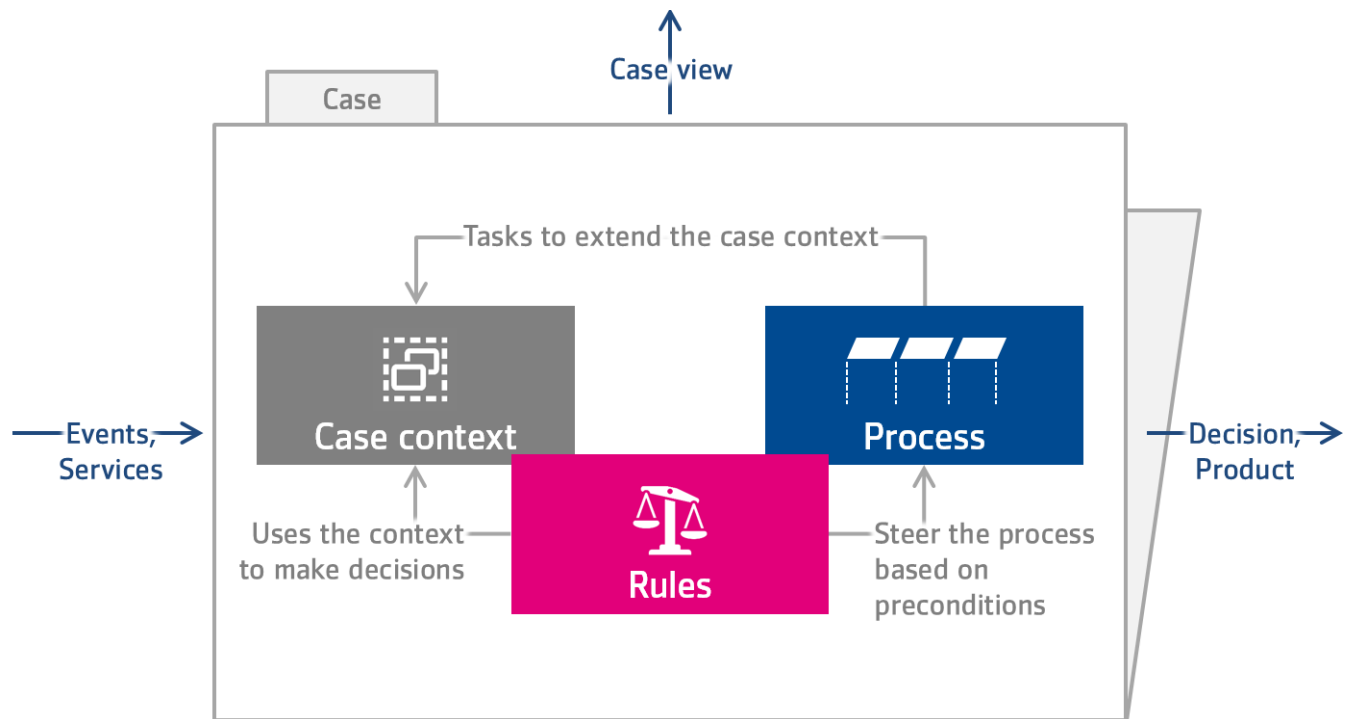
If a project contains more than one case, all entities of the table above need to be duplicated. It is preferred to create different projects for different cases, rather than prefixing or postfixing all entities (like Document_Application and Document_Complaint).

Relation between cases

When a case is related to another case, the related case "knows" with which case it is related, but not the other way around. The main case may use the result of the related cases, so related cases contribute to the main case. Furthermore, the fact that the main case does not know its related cases is also due to the fact that such related cases may come into existence after the main case is long closed. So a complaint case will contain the Id of the related application case (as part of the complaint case entity Case data), but the application case will not contain all Id's of the complaints that are related to it. Lists can be used in Blueriq to navigate from a main case into its related cases.

Characteristics of a case

As discussed in this article, a case the universe of Dynamic Case Management is more than just a process. Two other very important dimensions play a role in a case, namely Rules and the Case context. This is shown in the diagram below.

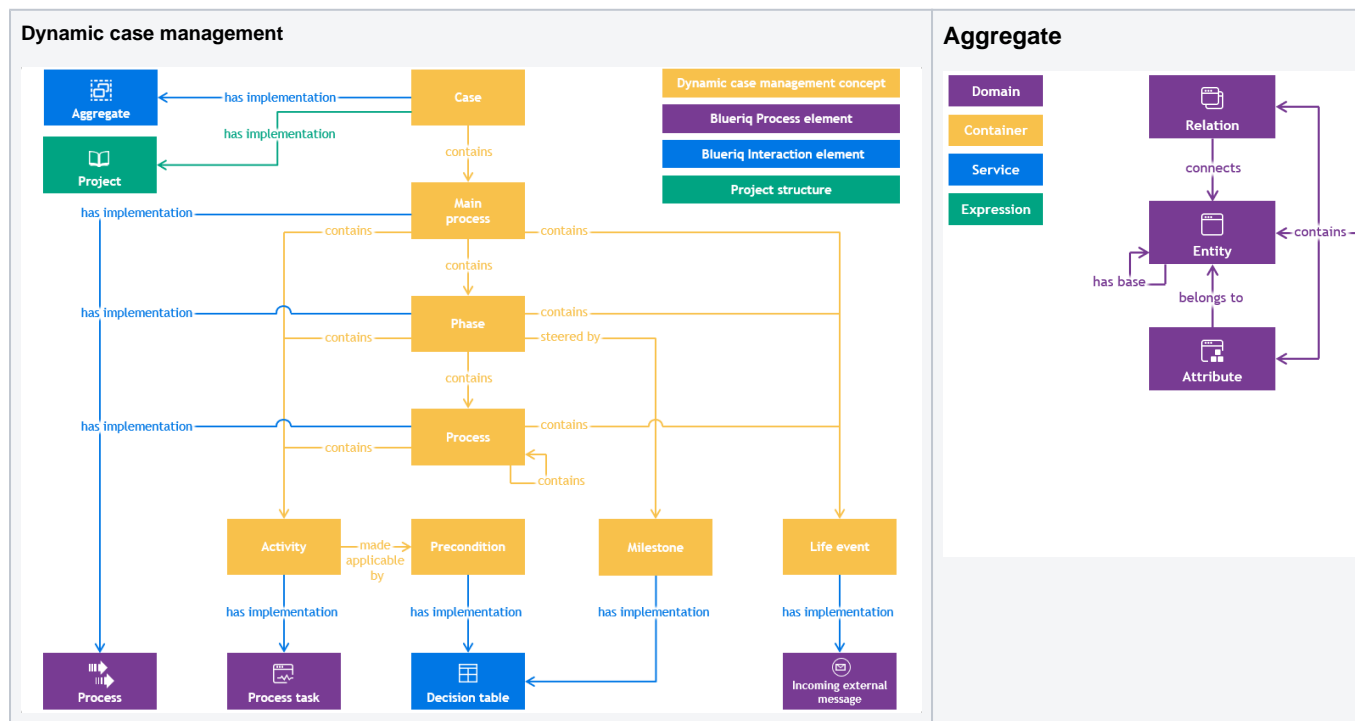


A case in a Dynamic Case Management design contains three main dimensions: Process, Rules and Case context. The **Process** is a set of activities that can or must be performed in order to reach a certain goal, for instance a payment, a permit or a benefit. The process in a dynamic case is not modeled with a predefined order in mind, using arrows. It contains independent activities, each with a precondition that states when the activity is relevant. These preconditions are **Rules** that use the **Case context**. This Case context is extended by input by users or services by means of activities of that process.

See also

More information on how to design dynamic processes in Blueriq can be found here: [Designing dynamic processes](#).
 Everything about persistency management and aggregates can be found here: [Persistency Management guide](#).
 A visualization of most concepts discussed in this article can be found here: [Blueriq visuals](#).
 A template for dynamic case management can be found here: [DCM Foundation - v4 \[editor\]](#) .

Specifically for dynamic case management and persistency management, see



¹ Case

In Blueriq a case is defined as an instance of a process. With the adoption of DCM and DCM's definition of a case as described in this document, there are two definitions of case present in the Blueriq Community documentation. At the moment, we are rectifying this.