

CARDS

Domain Event

Probably the most important element in Event Storming is the Domain Event. It represents a fact **something that has happened**. The corresponding color is **orange**. You can describe every activity, data flow or process as a series of Domain Events e.g. “*User Browsed Online Shop*” -> “*Item Put Into Cart*” -> “*Order Placed*” -> “*Order Paid*” -> “*Order Shipped*”. It’s this simple concept that makes Event Storming successful. A quick explanation how to formulate an Event and the Big Picture Event Storming session can start. **Use past tense in Domain Events**. Put them in chronological order from left to right to tell a story. Ask questions like: What other events need to happen before this event?, Who or what causes this event? When does the event happen?

Command

A command represents a decision that **something should happen**. The color code is **blue**. It can be a decision of a user e.g. who wants to *Place an Order* or *Pay an Invoice* as well as a decision based on a [Policy](#) or made in an [External System](#). Usually, commands are named with a verb in imperative mood, present tense - followed by subject e.g. “*Start Engine*”, “*Buy House*”, “*Share Picture*” ... Commands cause events. Often, you have a 1:1 relationship between them: *Pay Order* -> *Order Paid*, *Book Room* -> *Room Booked*. However, one command can cause one, multiple or no events. It depends on the [business rules](#).

Actor

Actor cards are used to highlight that users with a certain **role** are involved in a business process or use a specific [UI](#) to make [decisions](#). The color code is **lemon yellow** and the card is in portrait. Write the user role on the card and put it next to a UI or Command card.

Aggregate

The term aggregate is borrowed from Domain-Driven Design. In Event Storming it’s basically a business rule that makes sure a [Command](#) can be executed e.g. *An order can only be shipped if it is paid*. Write down such rules on **sunny yellow** cards and put them between a command and one or more events.

Information

Green cards represent information needed to make a [decision](#). Information are usually displayed in a [UI](#), but can also be in an excel sheet, PDF document or printed out on paper. [External Systems](#) fetch information via API or from a database to make decisions. All these cases can be visualized with green Information Cards in combination with others.

-

Every information change should be described as a [Domain Event](#). **If there was no event, the information did not change!** Developers familiar with the architecture pattern CQRS should also know the term “Read Model”. Both terms can be used interchangeably.

Policy

A Policy is a **reaction to an event**. It represents automated decision and coordination logic. Simply said: *If this, then that!*. **Lilac** Cards are used for policies. They trigger new [commands](#). A good example of a process with policies involved is a reservation. Imagine a ticketing system of an airline. While booking a ticket, one can usually select a free seat in the airplane. A resulting event could be *Seat Reserved*. As soon as a Policy receives the final event *Flight Ticket Bought*, it can trigger a command to change the *reserved seat* into a *booked seat*. This should happen within a fixed timeframe, let's say 15 minutes. If the potential passenger does not finish the booking process, the Policy will trigger a command to change the seat back to *available*.

Hot Spot

No Event Storming without Hot Spots. If you don't have a couple of **red** Cards on the board after Big Picture Event Storming, you did not dig deep enough. Just kidding, but a Hot Spot is the swiss army knife for an Event Storming facilitator.

Use Hot Spots when:

A question came up that nobody can answer right now -> Write it on a Hot Spot and put it close to the elements on the board that triggered the question. A discussion between two experts becomes heated or drifts into details so that no one else can follow anymore -> Stop the discussion, write the topic on a hot spot with a note that it should be revisited in a follow up. Someone points out a pain point in a process or describes something as a bottleneck -> Write it down and mark the area on the board with the Hot Spot. Hot Spots are good action items to tackle after an Event Storming session. They are the best tool in a collaboration session to keep focus. It's easy to zoom into process details or spontaneously discuss ideas with Event Storming. But at some point you'll need to stop to not lose focus on the main session topic. Hot Spots are made for such situations:

- Discovered a new bottleneck -> write it on a [Hot Spot](#)
- Found a promising new idea -> write it on a [Hot Spot](#)
- Need to address a bug in the system -> write it on a [Hot Spot](#)

To categorize Hot Spots you can use [Card Tags](#). We assign the tag **#later** to all Hot Spots containing ideas. Tagging Hot Spots allows you to filter them in the tree view. You can search for tags in combination with card types and also use a "NOT" operator to look up all Hot Spots that don't have a specific tag assigned.

External System

A third-party system involved in a business process can be visualized with a **dark pink** Card. Digitizing workflows often means connecting and orchestrating different digital services. Event Storming is a lightweight method to plan and document interactions between your system and external services.

UI / API

The original color code for UI stickies is white, but on prooph board all cards are white with a colored top bar. So we decided to invert the color and use **black** for UI Cards. We also extended its

meaning to include APIs. APIs as well as UIs are public interfaces. One for humans the other one for machines. To keep the color code simple, black Cards combine both - human and machine interfaces - to a unified **Interface Card Type**.

Use icons on UI Cards to roughly sketch what is visible on the screen. Just enough to support the idea. Don't waste your time with too much details. prooph board is not a wireframe tool. UX / UI designers have their specialised tools for this task. However, screenshots of wireframes / existing UIs are a quick and easy way to include more detailed designs in Event Storming sessions.

WHAT IS EVENT STORMING

Event Storming is a workshop-based method to collaboratively discover business processes and user flows. You can apply it practically on any technical or business domain, especially those that are large and complex. The fact that Event Storming puts events at the heart of the method makes it a perfect tool for event-based system engineering.

Purpose

Event Storming comes in different flavours which differ mainly in the level of detail. A typical starting point is a **Big Picture Event Storming**. It is recommended to invite all stakeholders such as developers, domain experts, decision makers, UX designers etc. to collect viewpoints from each participant. The group uses Domain Events written on sticky notes to explore the domain. By visualizing the entire business flow as a series of events, critical insights and discoveries are triggered and the foundation for an event-driven system is made.

How to get started with your first proophboard session. It's important to set expectations correctly and start small. A few pitfalls should be avoided to not risk success. Goal of the first session should be to a) get familiar with Event Storming itself and b) create a shared understanding of the domain or the area of the business that is of interest for you. Avoid goals like: After the session we should be able to derive stories that we can work on next. Such goals often require detail sessions. If you set them too early - maybe to convince a manager to spend time and money - you could end up with unmet expectations and unhappy attendees or supporters. Make sure to have a facilitator available. You need someone with Event Storming experience and moderator skills. It's possible to run out of time or end up in heated discussions, if you don't have someone who takes care of that. We can help you find a facilitator. [Just get in touch](#). Once you're familiar with the method, a team member can take over the facilitator role. Split a remote Big Picture workshop into at least two days. Each day not longer than 3 hours. Also plan enough breaks e.g. 10 min after every 50 minutes. Remote collaboration is fun but also draining! Keep that in mind. Nice side effect is, that you can give homework after the first day. At the end of day one ask questions like: *Are the events correct?*, *Did we forget something?*, *Should we change the current structure?*. Start day two with the same questions and see what new insights people got overnight.

Prepare Board Workspace

First of all you need to create a new board in prooph board. After sign in you can directly do this on the dashboard. Click “Add Board” to create a new one and give it a name. You’re redirected to the new, empty board. It’s ready to be prepared for the workshop session. People can be invited using the “share” button in the top bar. Enter their email addresses and they’ll receive an invitation.

First Event

The first event is the most difficult one. Everybody is still unsure what to do and how it works. But don’t worry, this changes quickly. A good starting point is to ask about the goals of the company/business unit. *What user needs do we serve? How do we make money*

Chaotic Mode

After the first event is on the board (the icebreaker event), you can ask every attendee to write down more events. This happens in parallel. Everybody should think about questions like: *What happens before the event? What comes after? What events are important for the company and our users?* Ordering is not important at this stage. Just eventstorm whatever comes to mind and put it somewhere. 10 - 15 minutes is a good timeframe for chaotic mode. Use prooph boards integrated timer to keep an eye on the clock. One can start a timer by clicking on the **hour glass** icon in the top bar. Set a time, press enter and the timer will start to tick. It is synchronized with all attendees. So everybody can see how much time is left.

-

Tell a Story

First Event. Now that we have a couple of high level Domain Events on the board, it’s time to order them chronologically. One person takes over the driver seat (does the ordering) and constructs a story. Switch driver seat every 5 - 10 minutes to keep people engaged. Talking about the process (the story) will trigger discussions. Better events are suggested, wrong or duplicate events get removed and more detailed events added.

Group By Context

The following steps can be mixed as needed. It’s the facilitator’s task to coordinate work. Meaning, it’s likely that you’ll jump between context grouping and domain exploration and identify Hot Spots along the way. Group events that belong together and try to find a name for the group. Again, someone should be in the driver seat, but change them every few minutes. If you’re a large group of participants, now is also a good moment to split into working groups. Each working group can explore one or more event groups.

Explore The Domain

It’s time to introduce more Event Storming concepts like [Information](#), [Actors](#), [External Systems](#) and [Commands](#). Zoom in by asking questions like *What needs to happen here exactly?* and use examples with personas and concrete data written on the stickies. This deep dive will unfold a lot of

new information and knowledge. This type of exploration needs a little bit practice. How much detail is too much for a Big Picture? When should we stop with the help of a Hot Spot and continue exploration in a follow up with a smaller group? Don't worry, you'll get better finding the right level of detail quickly.

Identify Hot Spots

[Hot Spots](#) are useful during the entire workshop. But still, if exploration get stuck a little you can push it forward again with explicit questions about bottlenecks or opportunities for improvements: *Is there an area where we could do better? What do our users feel about this user flow here? Should we look at this from a legal perspective?*

Retro

At the end of each day you should do a short retro. You can prepare 2-3 [Frames](#) with questions: *What do you like about Event Storming? What don't you like about Event Storming? Do you have open questions? Could the workshop be improved next time?* Ask participants to write down their answers on Cards and put them into the respective Frames. 5 minutes should be enough. Remember using prooph board's timer! Then go through the answers and let everybody explain their thoughts.

Feedback is always very important. It helps to learn and grow and give people a good feeling about the next steps.

What's next?

We hope you enjoyed your first Event Storming, and it was a success! Big Picture sessions are already very valuable, and you can repeat them or continue with specific areas of the value chain, zoom into sub processes and do detailed sessions. Every Event Storming will sharpen your knowledge of the domain and create a common understanding of the things going on in the company.

Event equals Information changed over time

We perceive our environment through events. If we see something, it already happened. The picture in our head is a picture of the past. As an example, if you look at a star in the sky at night you cannot really know if the star still exists. The light of the star needs a lot of light years to reach the earth and finally your eyes.

When we see, hear, smell, touch or taste something we basically process information. The difference between one set of information and another set of information makes our reality. Our senses process information changes that already happened aka **events**. **Events are at the core of our reality.**

Why does it matter?

When it comes to traditional software design, people tend to put a lot of focus on things and how they relate to each other. In other words, they focus on specific sets of information and how those sets are connected. That itself is not a bad idea. However, the order is wrong.

Information is always the result of one or more events.

If you design an information system without putting events at the core, you're doomed. You're ignoring the fundamental building blocks of our reality or at least give them not enough attention. This results in so-called accidental complexity. At the beginning of a new system everything looks clear and simple. But you continue to add more and more information changes and at some point you loose control over it. Without proper events, you cannot tell **why something happened**. Yesterday the system was all good then someone did something, and now it's broken, but you cannot perceive what happened, because you missed the events. Observability is very important as soon as the system reaches a certain level of complexity. Now one could argue: we have logs and logging tables for observability. We also have Mixpanel or Google Analytics in the frontend to track user behavior. So we know what happened. Well, that's maybe true. But again, the order is wrong. Usually logs in whatever way are an afterthought. They are not part of the design process, and they are not at the heart of the system. It's an additional layer instead of being the foundation. Which basically reduces efficiency, clarity and alignment.

Basic Pattern

You have Information. Changing the Information is an Event at a specific point in time. That gives you new Information, which can be changed again and so on. No matter how complex a system is as a whole, a single unit of work should always be designed around that simple pattern. And since events are at the core you can even simplify the pattern to this: Without an Event you cannot tell why Information is what it is at a specific point in time. It's just a snapshot like one frame of a movie or a thumbnail of YouTube video. A desired Change is also incomplete without an Event, because you cannot tell if the Change really happened. If you have an older snapshot of the Information, you can try it by comparing the old snapshot with the new one, but still you cannot be 100% sure that the desired Change was actually the one that caused the Information to change. Contrary to that, by looking at an Event you can say:

1. There was a desired Change, and it really happened because you can see the Event.
2. The Event produced a new set of Information.

So you can purely focus on events and design everything else around them. And that's the reason why a [Big Picture Event Storming](#) should always start with collecting events. It is one fundamental concept, and it is very easy to grasp. It provides you with everything you need to discuss behavior of a system, and it will be the same at any other scale or detail level.

Now lets look at a few different scales.

With User Interface

Information is viewed by a user in a user interface. The interface provides the user with an option to tell the system: “Please change the Information in a particular way!”. If and only if the system processes the desired Change a new Event has been occurred and the user can view the changed Information in the User Interface again.

With Digitized Rules

Good software systems don’t blindly trust the user. Usually they have business knowledge incorporated to protect invariants. What the user wants to change is one thing, what they are allowed to change is a different story. The best place for such rules is between a desired Change and an Event. If the Change is not accepted by the system, it is either rejected (so no Event happened at all) or a different Event occurs.

Event Modeling Overview:

Event Modeling is a method used to design and document event-driven systems by focusing on the sequence of events and the state changes they cause. It helps in understanding how a system behaves over time by mapping out the events, commands, state, and read models in a timeline format. Event Modeling provides a clear, chronological representation of how commands trigger events, how events change the system state, and how read models project these states for user queries. This helps in designing robust event-driven systems.

What can Cody do? Cody can translate your Event Designs into a working prototype with basic styling. This can be accessed by right clicking on the context frame of your feature and starting cody wizzard.

Cody Suggestions. [Cody](#) is deeply integrated on prooph board. It does not only translate designs into working code, but also helps you with the design itself. When [Event Modeling Mode](#) is enabled, Cody appears within a [Slice](#) to suggest cards. If Event Modeling is turned off, Cody switches the suggestion style to [Continuous Event Storming](#). The latter was the default format on prooph board, before we fully integrated Event Modeling. If you prefer this style, Cody will continue to help you with it.

Where to turn Off Cody Suggestions

You can disable Cody Suggestions in the board settings.

USE FRAMES FOR GROUPING

Frames are used to group [cards](#) and other elements. prooph board supports 3 frame types:

Context / Module Frame

Depending on the scope of the board, this frame refers to a team, a DDD Bounded Context, a sub-domain, a business capability, a software module or a (micro)service. It's meant to group information, concepts and design that belong together. And it's the next level in the [tree view](#) after layers.

Event Modeling Swimlanes

A Swimlanes Frame behaves similar to a Context, but ships with Event Modeling Swimlanes support. Each Frame contains a timeline, a user role lane and a system module lane. You can add more role and module lanes by clicking on the plus icon that appears when selecting the first or last lane label of the Swimlanes Frame.

Event Modeling Slice

Slice Frames split an Event Model into workflow steps. You can use them to group elements. Slices are the second level below contexts or swimlanes in the [tree view](#).

Bird View

When zooming out the content of a frame is collapsed and its label scaled. This provides you with a bird view to keep an overview on large boards.

The two frame types collapse at different zoom levels:

Context / Module: collapses later than **Feature / Slice** (later = more zoomed out).

The bird view can be disabled for a frame. Open the context menu (right mouse click / tap and hold on touch device) and toggle **Bird View** -> **Hide Details**. If you want turn off bird view for all frames, please see [Lite Mode](#).

Disabling the collapse functionality can be useful if you want to use frames as swim lanes or sections of a canvas.

Move Frames

To be able to navigate on the board even if a frame fills the entire screen, they only have a small drag zone. Otherwise you would always move the frame instead of moving around on the board.

As you can see in the GIF, the border of a frame gets highlighted when the mouse pointer is close to it. You need to select the frame, move your mouse next to border, so that the pointer turns into a hand and only then you're able to move the frame. Don't worry if it is a bit tricky at the beginning. You should get used to it quickly and it really helps to navigate on the board.

Change Shape

Frames can have different shapes e.g. to represent different kinds of modules or contexts.

Chose a shape from **context menu** -> **Style**

prooph boards comes with some handy keyboard shortcuts to quickly add, modify and arrange elements on a board.

Navigation

Keymap for Board Workspace

No Element Selected

Arrow Keys

Move around on the workspace (hold for faster movement)

Shift + Arrow Keys

Move fast on the workspace (hold for even faster movement)

Ctrl/Cmd + +/- (Numpad)

Zoom in/out

Ctrl/Cmd + F

Focus search in sidebar

Ctrl/Cmd + V

Paste copied or cut elements

Tab

Select first element in the viewport

Ctrl/Cmd + Shift + Backspace

Go back to the previous position on the board

Single Element Selected

Arrow Keys

Move element (hold for faster movement)

Shift + Arrow Keys

Move element fast (hold for even faster movement)

Ctrl/Cmd + Arrow Keys

Resize element (hold for bigger resize steps)

Ctrl/Cmd + Shift + Arrow Keys

Resize element with bigger resize steps

Ctrl/Cmd + F

Search similar elements in the sidebar

Ctrl/Cmd + M

Show metadata of element

Ctrl/Cmd + G

Trigger Cody with element

Ctrl/Cmd + C

Copy selection

Tab

Select next element

Del/Backspace

Delete element

Enter

Edit element label

Esc

Deselect element

Alt + E

Change sticky type to Event

Alt + C

Change sticky type to Command

Alt + A

Change sticky type to Aggregate

Alt + R

Change sticky type to Role

Alt + D

Change sticky type to Document

Alt + S

Change sticky type to External System

Alt + H

Change sticky type to Hot Spot

Alt + P

Change sticky type to Policy

Alt + U

Change sticky type to UI/API

Ctrl/Cmd + Shift + F

Move to front

Ctrl/Cmd + Shift + B

Move to back

Alt + Shift + Page Up

Select parent element

Alt + Shift + Page Down

Select first child element

Alt + Shift + Arrow Key

Copy and connect selected element

Ctrl/Cmd + Enter

Duplicate element

Multiple Elements Selected**Arrow Keys**

Move elements (hold for faster movement)

Shift + Arrow Keys

Move elements fast (hold for even faster movement)

Esc

Deselect elements

Ctrl/Cmd + G

Trigger Cody with elements

Ctrl/Cmd + Shift + L

Align elements left

Ctrl/Cmd + Shift + R

Align elements right

Ctrl/Cmd + Shift + C

Align elements center

Ctrl/Cmd + Shift + U

Align elements up

Ctrl/Cmd + Shift + D

Align elements down

Ctrl/Cmd + Shift + M

Align elements middle

Text Editing (Element Label)

Esc

Finish editing

Ctrl/Cmd + Shift + L

Insert link

Ctrl/Cmd + Shift + H

Insert horizontal ruler

LITE MODE

If you encounter performance problems on large boards or during video calls, you can turn on the Lite Mode. This will disable zoom effects like hiding elements or scaling the font size of Bounded Context and Feature frames.

Effects turned off means your browser has less expensive tasks to do. This saves CPU and therefore also battery time.

Disadvantage of course is, that you no longer get the high level overview. Use the navigation tree and search in the left sidebar to still find the information you're looking for.

Switching between Normal and Lite Mode can be done in the workspace top menu. It's the battery icon next to your avatar. The following table compares both modes:

The chosen mode is persisted across sessions. If you prefer to always work in Lite Mode, you can set it once.

prooph board offers 3 different levels of setting access rights for boards. This can be done on the board settings page that you can either reach from the board overview (by clicking on the edit button of a board) or from the top menu on a board workspace.

Please note: Only board owners and organization admins can access board settings. If you don't have this role, you won't see the edit button or settings entry in the top menu.

The 3 Levels can be found in the Permissions section:

User Level

You can set **read** and **write** access for a specific user on this level. An input field allows you to look up members of the organization by username or email. If a member is found your settings become active right away.

If the person is not part of your organization, you can invite them by providing their email address. In this case the person will receive an invitation email with an activation link.

Access will be granted only if the invited person clicks on the activation link in the email and if you have enough user quota available in the organization. The invited person will be registered as a guest in the organization. For more information see [Guests](#)

Open invitations are also listed in the Permissions section. You can resend an invitation email or revoke an open invitation.

In the top bar of a board workspace you'll find a **share** button (again, only if you're owner or admin). This button gives you quick access to user level permissions. If you forgot to invite someone, you can quickly do it right there.

Team Level

read and **write** access to a board can be granted to an entire team. Select one from the dropdown and click on the **share** button. Everyone in the team will now have access.

All (Others) Level

On the **All** tab you can set organization-wide permissions or even allow anyone to access your board.

While user and team permissions will also make the board visible on the board overview for every person, this is not the case for **All** permissions. So people need to know the direct link to the board.

Access for anyone works as long as enough user quota is available in the organization. Each unknown person who opens a board is added as [guest](#).

With an organization you can share your board and user quota with colleagues. They don't need to worry about limitations, can easily create boards and share them with members.

Create an Organization

Each account can have exactly one organization. It's connected with the selected payment plan and therefore with available board and user quota.

To create an organization navigate to "Organization" management via sidebar, click "Create Organization" and give it a name.

Invite Members

Click on the blue user button next to your avatar. A dialog will open asking you to enter the email address of the member you want to invite. They'll receive an email with an invitation link. If they're not registered yet, they can do it for free. Once a member has joined an organization, they can create boards as long as the organization has enough quota.

Promote Member to Admin

Promoting members to organization administrators helps you to distribute administrative tasks like cleaning up no longer used boards, invite or remove members and more.

An administrator can:

- Invite or remove members
- Has access to all organization boards
- Has access to all organization teams
- Can manage access rights
- Promote a guest to member

Only the organization owner can:

- Promote/demote administrators
- Change payment plan

Member Lookup

Boards can be shared and people can be invited to join teams. Within an organization this is very easy. When asked for an email address just start typing to get suggestions for members. You can lookup email addresses and names.

Guests

It's also possible to invite external people to organization boards and teams. They will only get access to the shared board or in case of a team to all team boards. A guest cannot create additional boards within the organization context. And a guest has no access to the member list.

You can promote guests to organization members.

Migration

You've created an organization and now want to move boards and teams to it? No problem, switch to your private workspace using the workspace select in the sidebar below the organization item. You'll find an option to move your boards and teams to an organization on the "Board Edit" and "Team Management" pages.