

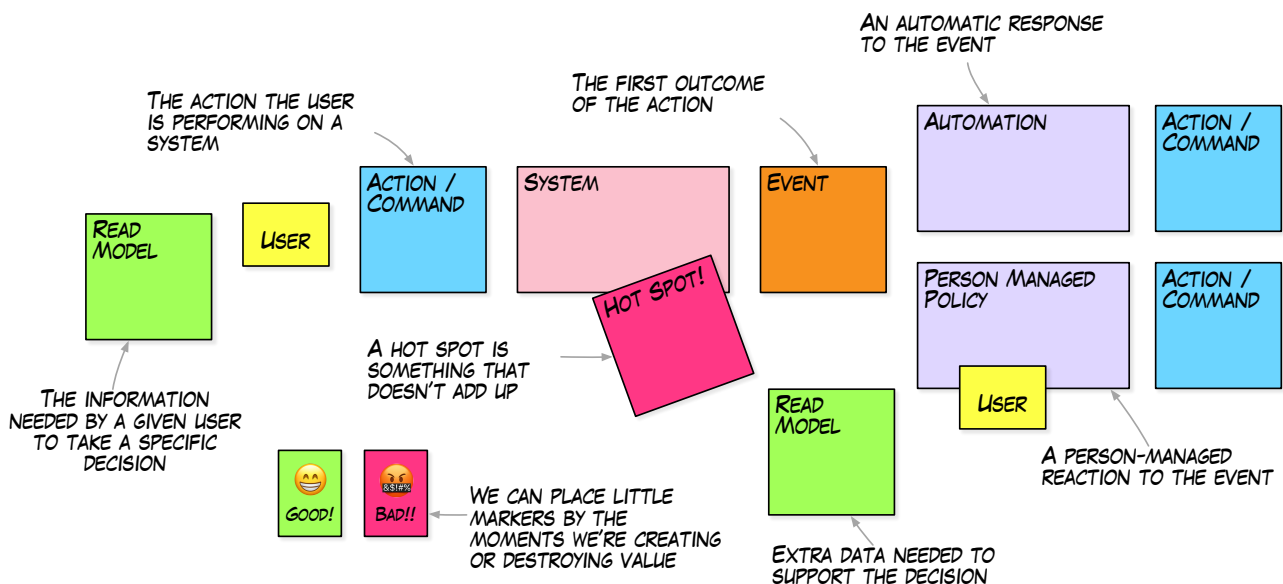
Rules of the game

A modelling session is like a collaborative game. Rules are fixed, but the strategy is up to you.

Game Rules for Process Modelling EventStorming

- 1) Every process might **end in a stable state**, usually a combination of an Event and a Read Model.
- 2) The **colour grammar** must be respected.
- 3) Every involved stakeholder should be **reasonably happy** (using value stickies to visualise it).
- 4) Every **hotspot** that should arise during the modelling session should be addressed.

And the mandatory colour grammar:

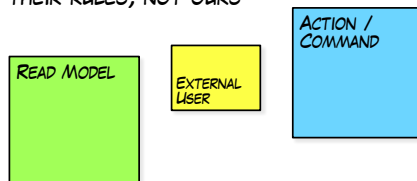


The event may contain all we need...



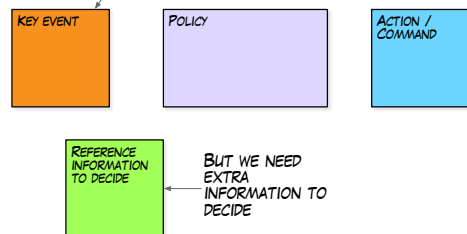
No policy for external users:

WE DON'T USUALLY NEED A POLICY FOR EXTERNAL USERS: THEY ARE FOLLOWING THEIR RULES, NOT OURS



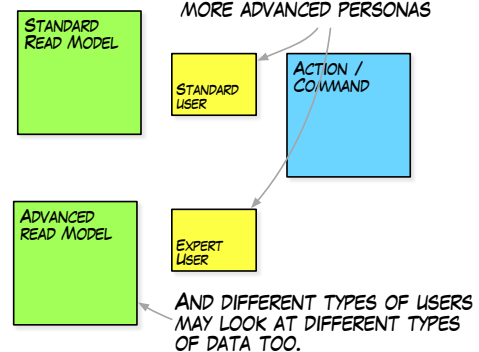
Or we may need the support of a read model:

THE EVENT MAY ACT AS A TRIGGER...



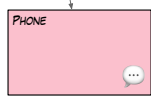
Specific Personas:

HOWEVER, WE MIGHT BE MORE SOPHISTICATED AND MODEL WITH MORE ADVANCED PERSONAS



Conversational Systems:

MANY THINGS CAN HAPPEN IN CONVERSATIONAL SYSTEMS, OFTEN AT THE SAME TIME.

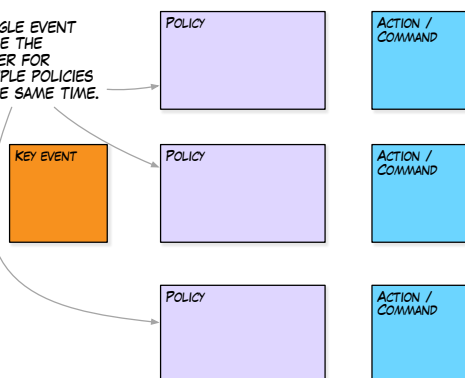


LOOK FOR THE TERMINATION EVENT OF THE WHOLE CONVERSATION: WHAT CAN MAKE IT END?

... BUT THE TERMINATION EVENT CAN BE TOO GENERIC TO BE ACTIONABLE. MORE SPECIFIC VERSIONS CAN SERVE AS TRIGGERS FOR YOUR POLICY.

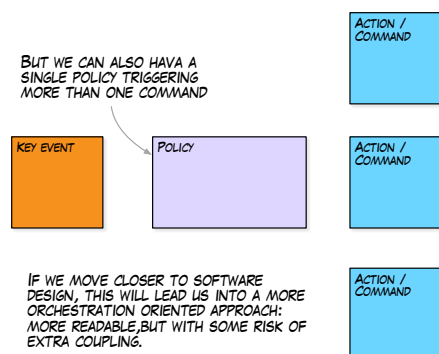
One Event, multiple policies.

A SINGLE EVENT CAN BE THE TRIGGER FOR MULTIPLE POLICIES AT THE SAME TIME.



One policy, multiple commands.

BUT WE CAN ALSO HAVE A SINGLE POLICY TRIGGERING MORE THAN ONE COMMAND



IF WE MOVE CLOSER TO SOFTWARE DESIGN, THIS WILL LEAD US INTO A MORE ORCHESTRATION ORIENTED APPROACH: MORE READABLE, BUT WITH SOME RISK OF EXTRA COUPLING.