**📜 Specification: Declarative Interactive Scenario Engine**

**🧠 Core Philosophy**

Design a flexible system for building interactive, multimedia, and behavior-rich experiences where:

* **Content lives in a CMS (Sanity)**
* **Behavior is orchestrated by dynamic Scenarios ("Scripts")**
* **User interaction can alter the flow**, including LLM-guided paths
* **Custom transitions and animations are declarative but extensible**
* **Device context (mobile/desktop) drives how navigation occurs**

**📂 1. Data Architecture**

**🔹 1.1 Card Schema (Sanity)**

Each **Card** is a self-contained content unit:

{

name: 'card',

type: 'document',

fields: [

{ name: 'title', type: 'string' },

{ name: 'cardType', type: 'string' },

{ name: 'localizedContent', type: 'object' },

{ name: 'componentName', type: 'string' }, // Optional custom React component

{ name: 'props', type: 'object' },

{ name: 'transitions', type: 'object' }, // enter/exit behavior

]

}

**🔹 1.2 Script Schema (Sanity)**

A **Script** defines a *scenario*:

{

name: 'script',

type: 'document',

fields: [

{ name: 'title', type: 'string' },

{

name: 'sequence',

type: 'array',

of: [

{

type: 'object',

fields: [

{ name: 'card', type: 'reference', to: [{ type: 'card' }] },

{ name: 'navigation', type: 'object' },

{ name: 'customAnimation', type: 'string' },

]

}

]

}

]

}

**🔸 Navigation Object**

Defines how and when to move to the next card or script:

{

onClick: { goToCardId: "card-2" },

onTimeout: { duration: 5, goToCardId: "card-3" },

onChoice: {

calm: "card-calm-1",

intense: "card-intense-1"

},

onLLM: {

promptTemplate: "Based on user's input and past choices: {{ history }}",

resolver: "choosePath" // hooks into LLM logic

},

onScriptEnd: {

goToScriptId: "script-2"

}

}

**⚙️ 2. System Components**

CMS (Sanity)

└── Cards

└── Scripts

Frontend App

├── ScenarioManager

│ ├── Loads script and card data

│ ├── Tracks current step, handles transitions

│ └── Loads new script if required

│

├── NavigationEngine

│ ├── Listens for events (click, swipe, timeout, etc.)

│ ├── Executes transitions

│ └── Positions cards based on transition type

│

├── CardFactory

│ └── Renders standard/custom React components from card data

│

├── InteractionMemory

│ └── Stores user inputs, choices, and history

│

└── LLMResolver

└── Calls LLM API to determine next card/script

**🧠 3. Navigation and Layout**

* The NavigationEngine is responsible for **spatially arranging cards**
  + Horizontal stack for slideLeft, slideRight
  + Vertical stack for slideUp, slideDown
  + Absolute overlays for fade, maskedReveal
* The engine determines layout **per transition type**

CurrentCard ───► NextCard

[x: 0%] [x: 100%] → slideLeft

[y: 0%] [y: 100%] → slideUp

[z: 10] [z: 0] → fade

**🧩 4. Card Types**

**✅ Standard Cards**

* TextCard, VideoCard, ImageCard, 360VideoCard
* Controlled transitions, layout-aware

**✳️ Custom Cards**

* Specified by componentName
* Full freedom to implement layout, transitions, and logic
* Trigger completion via callback

**🔁 5. Dynamic Transitions**

* Cards can specify customAnimation string (CSS class, Framer variant, etc.)
* Cards can be stacked or positioned in containers that enable the visual effect
* Transitions are triggered by the NavigationEngine

**🤖 6. LLM Integration**

* Used when next step depends on user history/input

onLLM: {

prompt: "User said: {{ input }}. Past: {{ history }}. What card comes next?",

model: "gpt-4",

resolver: "choosePath"

}

* Maintains interactionMemory:

{

inputs: [...],

choices: [...],

cardHistory: ["card-1", "card-2"]

}

**🌐 7. Device Adaptivity**

* ScenarioManager detects device type
  + Mobile → swipe gestures
  + Desktop → scroll and keyboard
* Layouts and transitions are rendered accordingly

**🧪 8. Validation & Future Considerations**

* Preflight validation:
  + Check all card references exist in Sanity
  + Check componentName exists in component map
* Future:
  + Branching conditions
  + Preview mode
  + Accessibility (reduced motion, screen reader support)