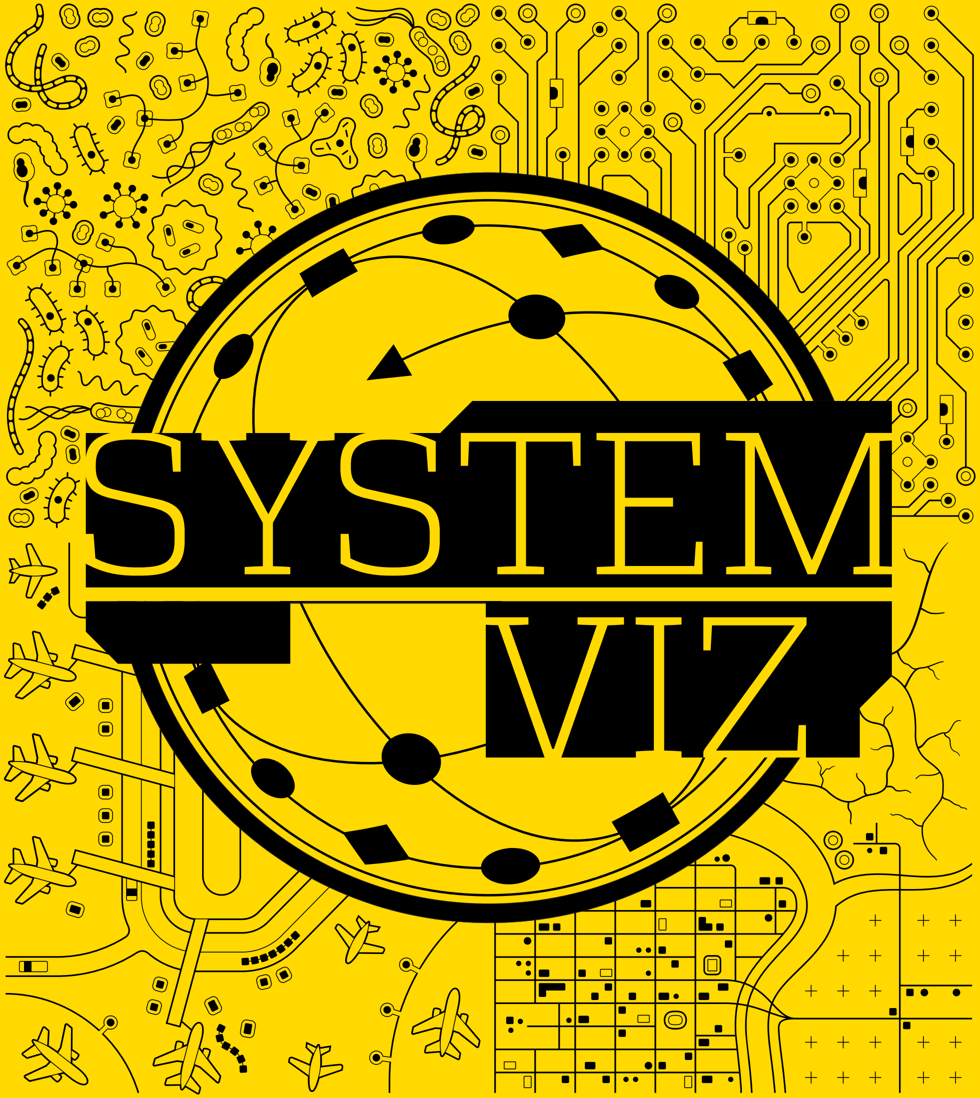


making health systems transparent with the **escalade** information-design framework

PETER STOYKO

19TH IIID VISIONPLUS CONFERENCE
VIENNA, AUSTRIA

2 5 · 0 5 · 2 3

SystemViz is an ongoing research-and-design project that explores better ways of visualizing systems. The concern is that system-visualization practices have not kept pace with the complexity of systems and their entanglement in our lives. As new media emerge (such as immersive video games and mobile touch interfaces), we get tantalizing glimpses at the potential of technology to represent systems in vivid and insightful ways.

However, much of that potential goes unrealized. SystemViz is an attempt to remedy the situation. A couple of practical tools have been released so far, notably the *Visual Vocabulary of Systems*, featured in the *International Institute for Information Design* (IIID) *Awards* this year (Gold/Distinction). I am here today to propose another tool that would benefit from feedback by the information-design community.

new

ESCALADE

INFORMATION DESIGN FRAMEWORK FOR SYSTEM MAPPING

This presentation introduces ***Escalade***, an information-design framework for mapping systems. The framework offers guidance for improving the way we depict complex systems of whatever kind. The intention is to ultimately create a new diagramming notation that (a.) can express the full range of system elements and dynamics (b.) using emerging forms of interactive media to (c.) make system analysis more accessible to everyone.

Escalade is an important stepping stone towards that goal. It can also be used to improve existing diagramming and mapping methods. Indeed, the shortcomings of these methods will be itemized before the framework is explained in full. Given that healthcare is the theme of this conference, examples from health systems will be used. Before getting to any of that, what's in a name? Why is the framework called *Escalade*?

analogy

The word "escalade" comes from castle siegecraft. It refers to the ladders used by invaders to climb castle walls. An **analogy** is being drawn. It is difficult to see the contours of many systems because they are hidden from view, much like the walls of a castle hide what is inside. Escalades are what enable a peak over the wall. Similarly, the information-design framework *Escalade* is a scaffolding for making comprehensible the various bits and pieces found within system maps. It enables a peak into how systems work from on-high. In some cases, systems are made deliberately opaque by those who benefit self-servingly from obscurantism, much like the castle defenders benefit from the opaqueness of solid walls. *Escalade* is an attempt to level the battlefield, so to speak, by making systems interpretable to an audience beyond insiders.

mnemonic

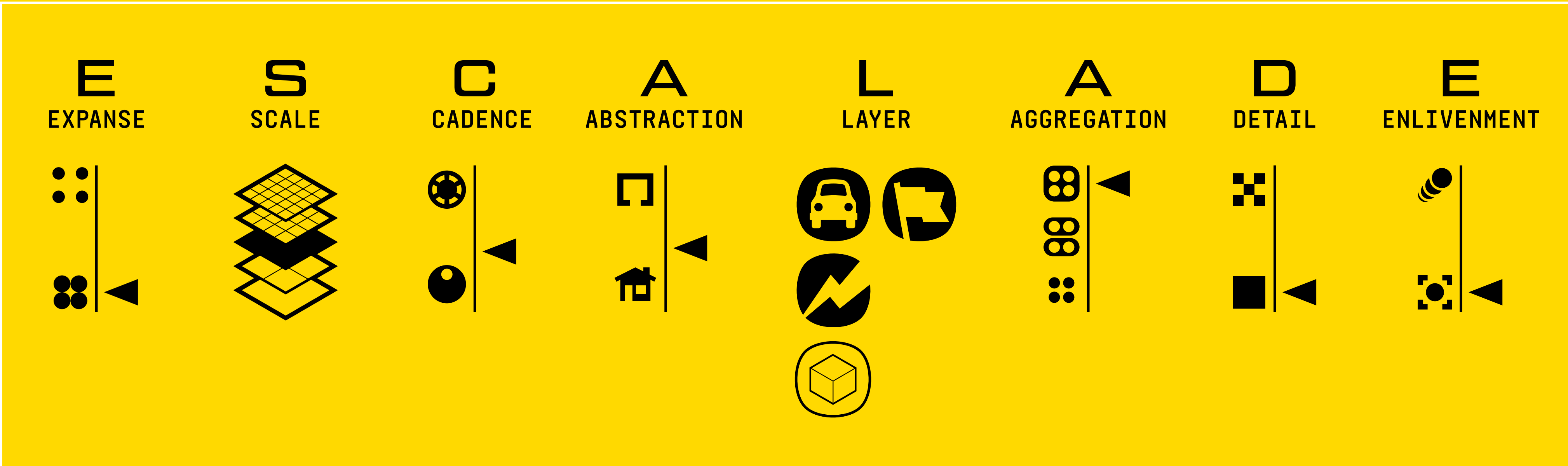
ESCALADE

EXPANSE / SCALE / CADENCE / ABSTRACTION / LAYER / AGGREGATION / DETAIL / ENLIVENMENT

The name is also an acronym—a **mnemonic** device for remembering the components of the framework. Eight dimensions of system mapping are kept in mind during design. A few relate to systems specifically, a subject with particular characteristics that are a challenge to describe, either visually or textually. The rest are about the cognitive challenges of how humans absorb visual information about things they have limited first-hand experience with, turning those interpretations into useful mental models. We should aspire to create graphics that are inviting and intuitive to a diverse audience. Any graphic will frustrate if it fails to accommodate all-too-human limitations and lack of knowledge. *Escalade* helps map designers think about that. Instead of treating these dimensions as a list, however, think of them more like a control dashboard …

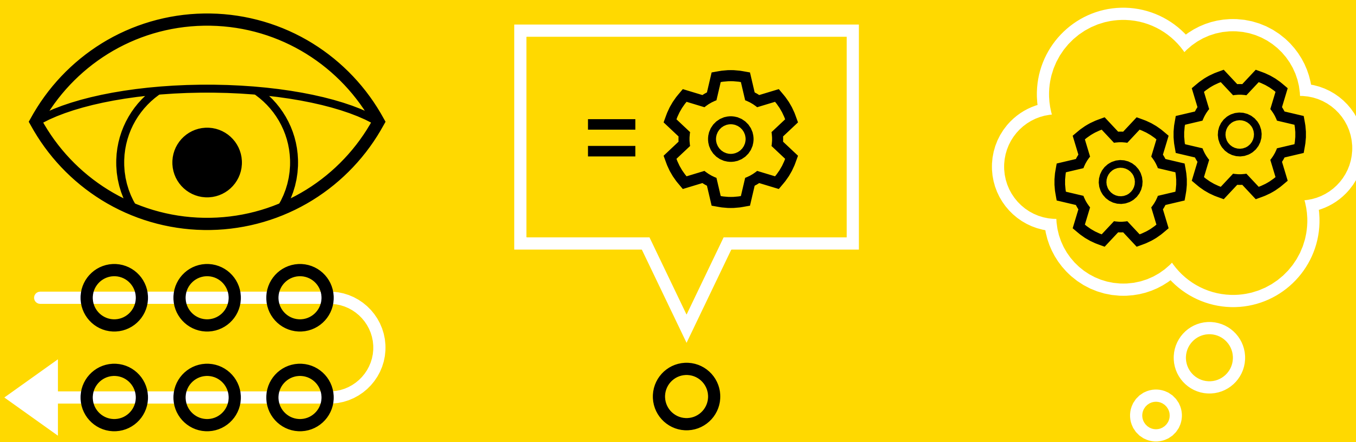
framework

DASHBOARD



Most dimensions are continuums that can be "dialled up" or "dialled down" to suit the usage scenario. Icons indicate the poles of each continuum. How the dashboard works will be explained soon. For now, it is enough to review each **dimension** briefly. *Expanse* refers to the treatment of spatial distance and size of objects in the map; that is, the extending and collapsing of space. *Scale* is the level of analysis: bird's eye view, microscopic view, and so forth. Cadence refers to time: from static snapshot to all temporal rhythms included. *Abstraction* is the extent to which objects are depicted accurately or using some sort of symbolism. There is a *layer* for each type of system shown. *Aggregation* refers to how like items are grouped into kinds. *Detail* is the amount of visual- and textual distinction. *Enlivenment* is the inclusion of motion, sound, and interaction.

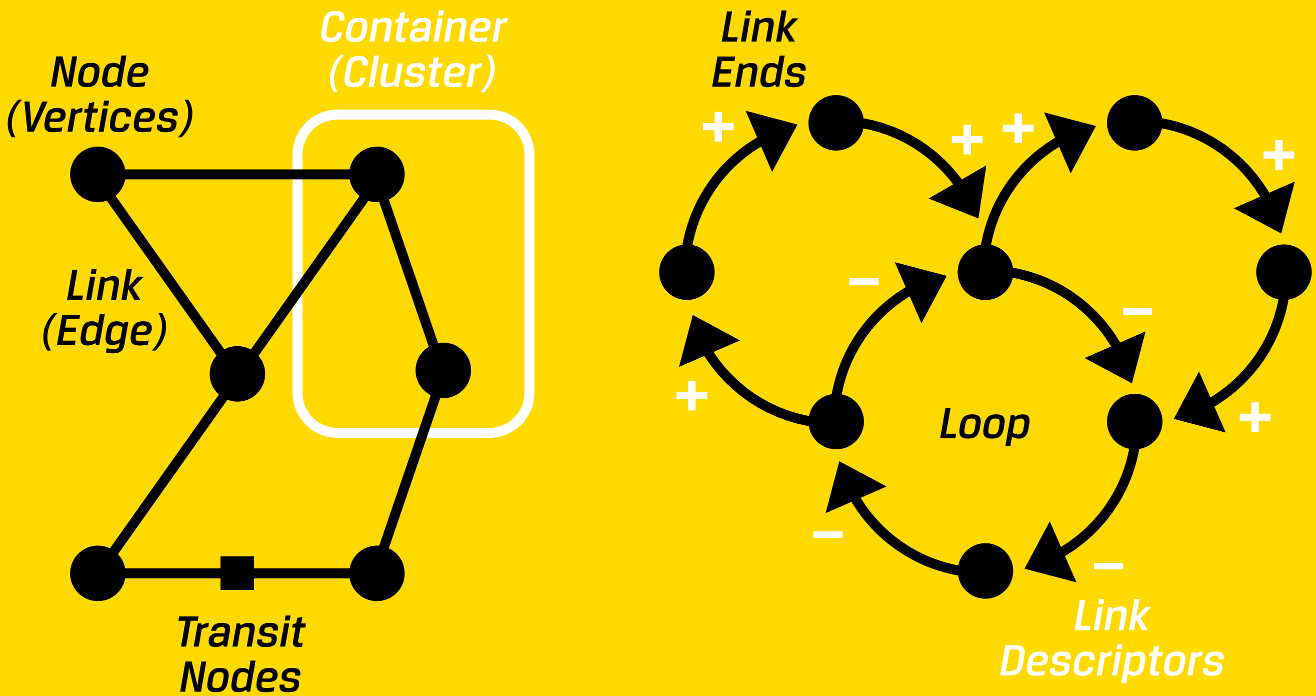
viewer tasks



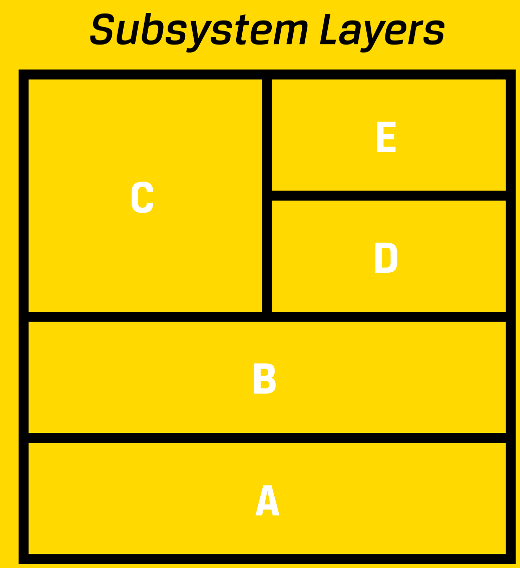
PARSE INTERPRET REASON APPLY

To lay the groundwork for evaluating the framework, consider the **tasks** performed by the viewer. First, the viewer has to *parse* the system map efficiently, identifying points of interest and situating them within a larger picture. Next, the viewer *interprets* the meaning of each item. Some initial learning may be necessary but interpretation should not require strenuous effort. The viewer then *reasons* about the system. A limited number of ideas can be juggled in the mind at once. Accordingly, the map should not overly tax concentration and memory. Noteworthy connections should be easy to grasp. Finally, the system map is part of a larger usage scenario. *Applying* findings to other practical tasks should be straightforward. If we look at popular system mapping methods, it is not hard to see how they fall short on these tasks. Let's review a few chronic problems.

basic forms



NETWORK DIAGRAMS (HYPERGRAPHS)

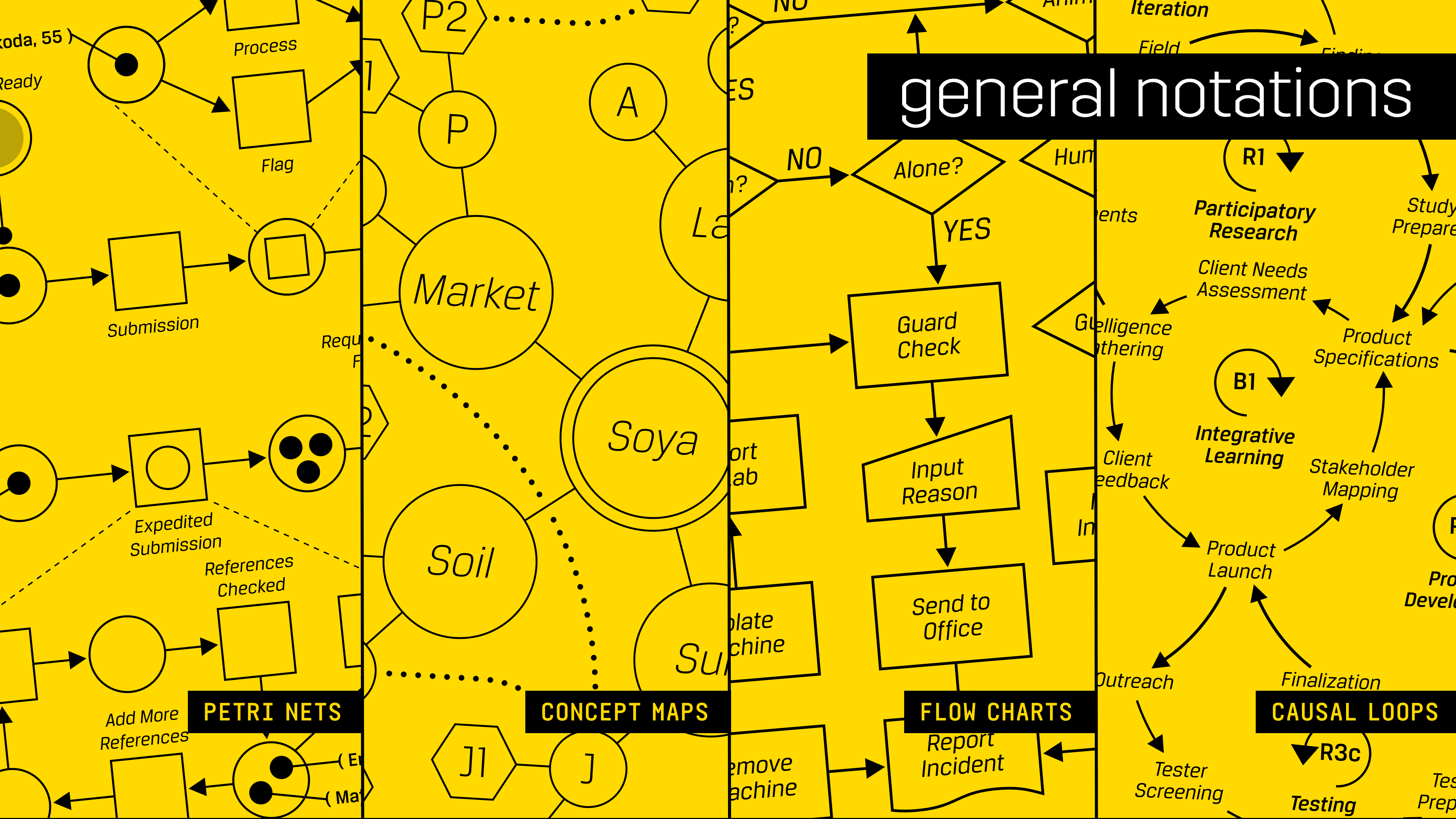


SYSTEM STACKS

As Donella Meadows puts it, "A system is a set of things—people, cells, molecules, or whatever—interconnected in such a way that they produce their own pattern of behavior over time."* Thus, in its **basic form**, a system diagram shows relationships between interacting parts to reveal a larger whole. That is usually done spatially with a network diagram (*hypergraph* in graph theory) containing nodes (vertices) and links (edges). Links denote relations, such as lines with arrows indicating causal influence and directionality. Text labels spell out what the parts represent. Systems can also be shown as abutting blocks (or "stacks") of subsystems using tables, with vertical positioning implying different levels of analysis or orders of functionality.

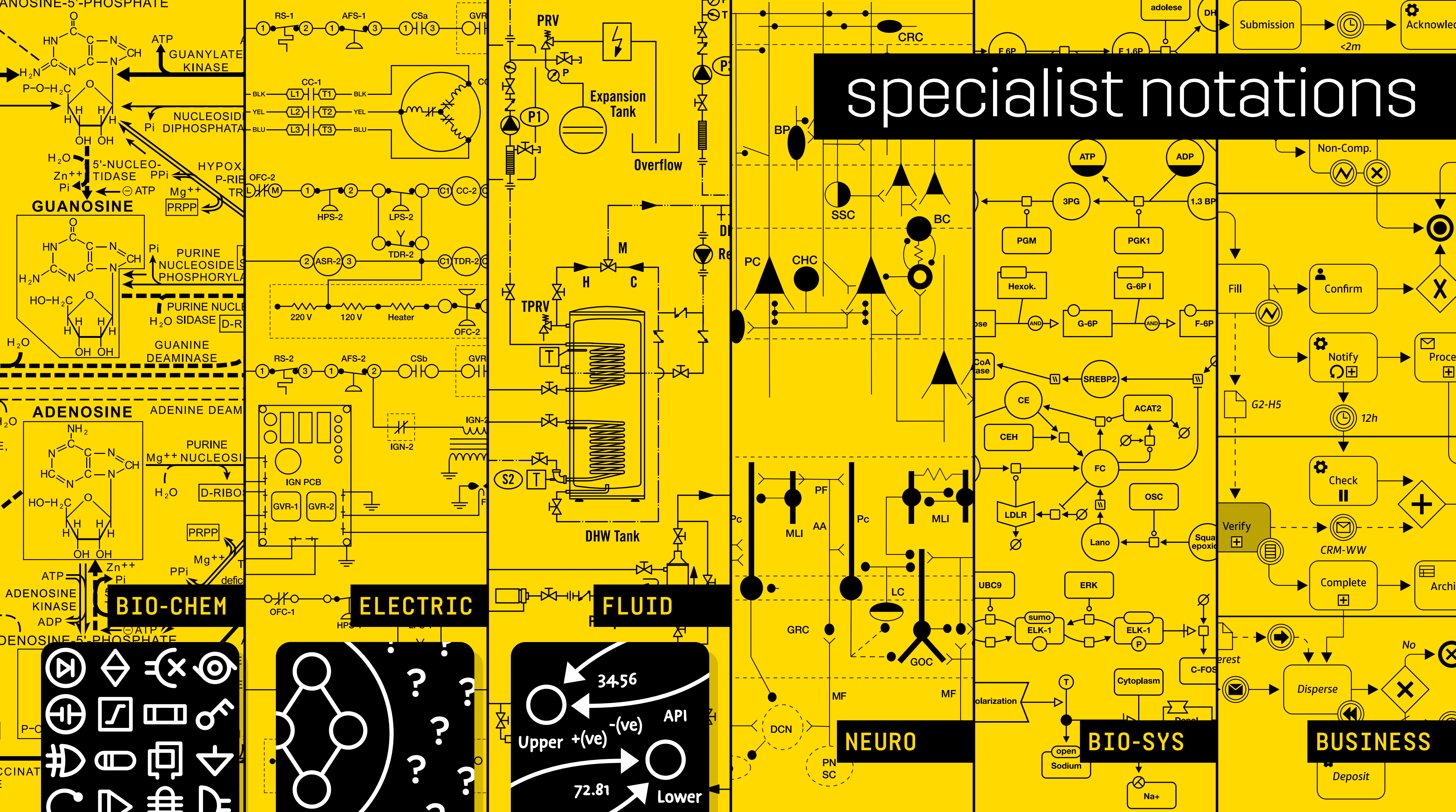
* Donella H. Meadows, *Thinking in Systems: A Primer* (White River Junction, VT: Chelsea Green Publishing, 2008), p. 1.

general notations



Many **general notations** have emerged from those basic forms to elaborate on system workings. For example, *petri nets* are node and link diagrams that make sequence and state changes more explicit by incorporating counters (●) that move between stations (nodes). *Flow charts* use nodes of different shapes to describe processes, turning them into verbs within a larger branch logic represented by arrows. There are also various flavors of *causal loop diagram* that show causal chains or sequences of processes, including how non-linear dynamics branch into and out of recurring loops of interaction. Every notation has an implicit theory of what a system is and how it works, making it awkward if different premises are adopted. General notations offer flexibility at the cost of specificity, which has lead to the proliferation of another family of notations.

specialist notations



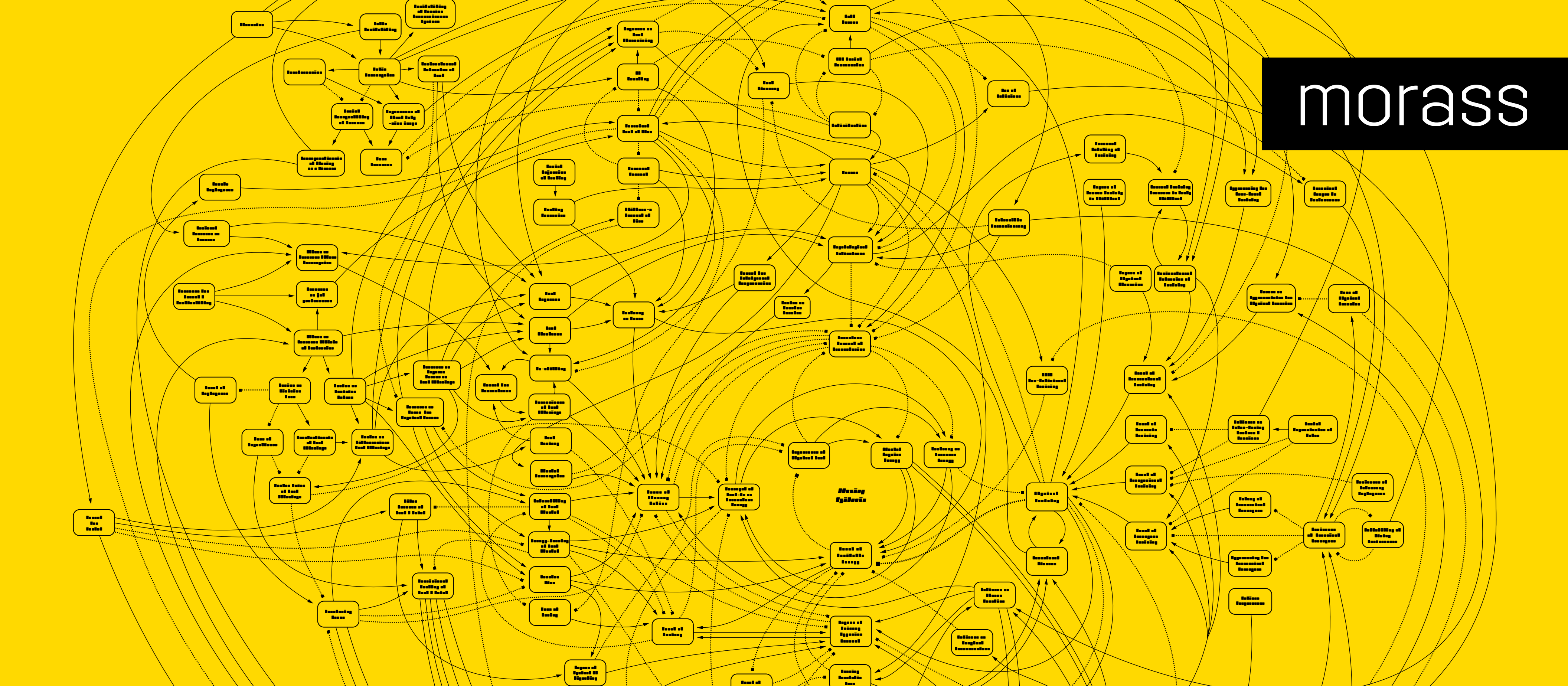
SYMBOL-ENCODING OVERLOAD
Too many encoded symbols have to be memorized to interpret the graphic

EXOGENY BLIND-SPOT
System parts not easily encoded are treated as out-side of scope or unknowable

THE ATTACHMENT PROBLEM
Free-floating descriptors lose an obvious connection to items they refer to

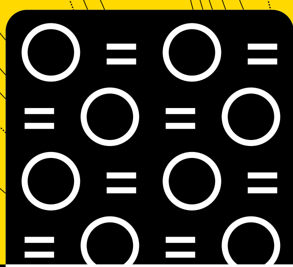
Technical distinctions are encoded into **specialist notations**, often using cryptic symbols and acronyms that take a long time to learn and make it hard for novice viewers to interpret. Free-floating descriptors are not always easy to attribute to particular items, especially in dense maps. The biggest shortcoming of specialist notations is the disciplinary boundaries they reinforce. System thinking is about showing how things mesh together. Yet these notations are about myopically cleaving reality into thin slices for inspection. Everything else is treated as *exogenous*; that is, as an inexplicable outside influence.

morass



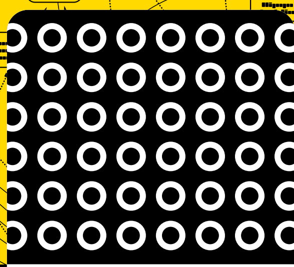
THE SPAGHETTI PROBLEM

A jumble of undifferentiated, overlapping links make the graphic hard to parse



FALSE EQUIVALENCY

Items of greater and lesser importance treated as being equal graphically

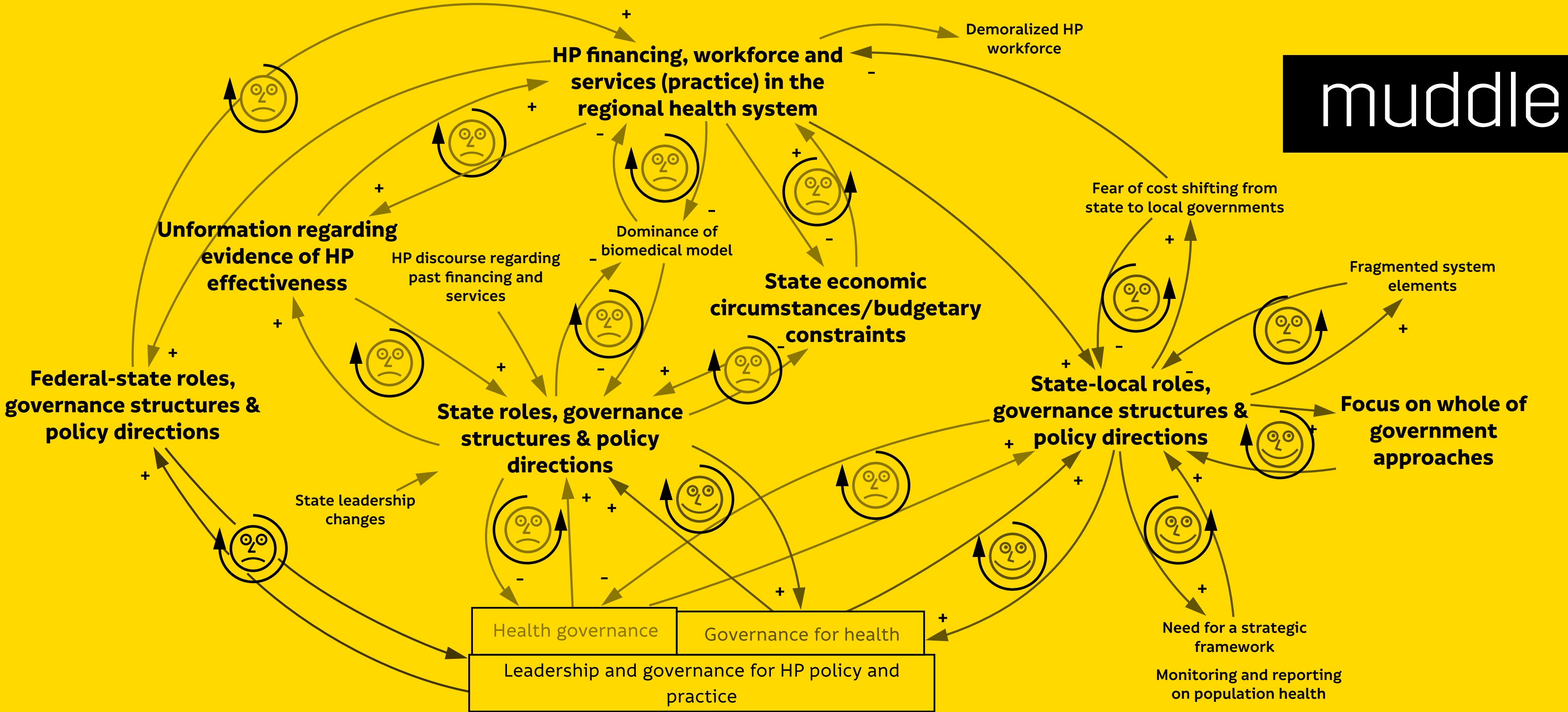


THE DEUTSCH LIMIT

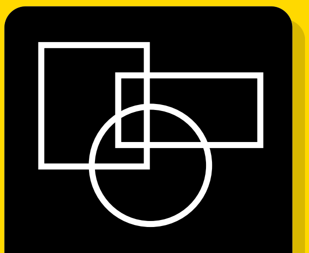
Salient parts of complicated systems outnumber visual items that can fit a display

The criss-crossing links of **large maps** look like untidy tangles of spaghetti with poor sight-lines. With general notations, each item has little visual differentiation, giving the overall impression that everything is more-or-less of equal importance. Any complicated system will have an abundance of parts that cannot all fit on a regular-sized display while staying legible. (The computer scientist Fred Larkin dubbed that *The Deutsch Limit*, named after L. Peter Deutsch who made the arguable claim that 50 nodes is the maximum capacity of most screens.) Few viewers are willing to persevere through the morass.

muddle

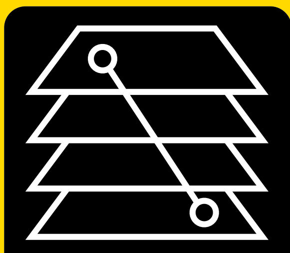


Source: Littlejohns et al, 2018.



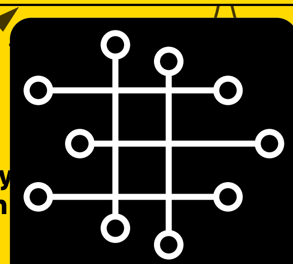
THE ABSTRACTION TRAP

Use of simple shapes + high concepts make real-world analogues hard to imagine



LEVEL CONFLATION

Items operating on different orders of magnitude shown interacting straightforwardly



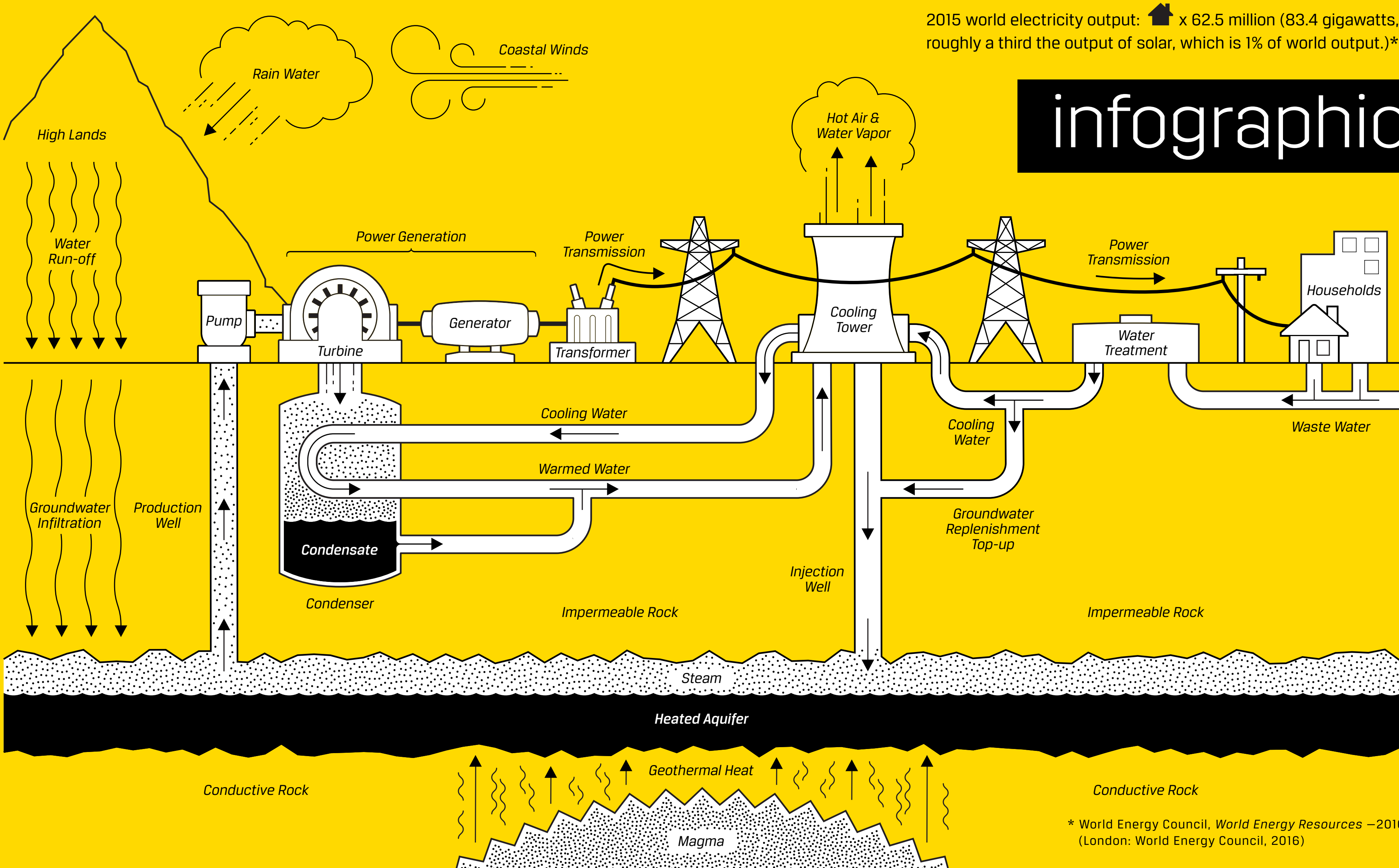
RELATIONAL AMBIGUITY

Links between items are unlabeled or otherwise under-described graphically

Mapping with simple shapes, ambiguous concepts, and unlabel-led links creates woefully **superficial models**.* Viewers struggle to think past the abstraction to imagine the underlying reality. It is too easy to draw visual non-sequiturs and causal leaps across orders of magnitude; the intermediate links in the causal chain are left as assumptions. Unlabeled (or otherwise under-specified) links make it unclear what the relations are about.

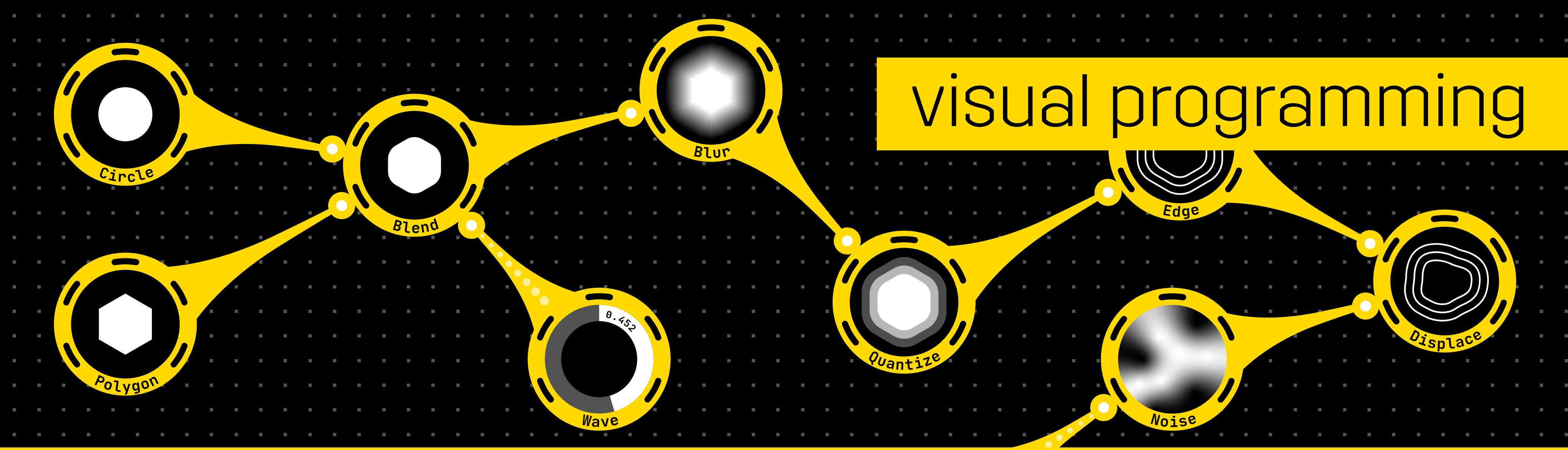
* Graphic above from: Lori Baught Littlejohns et al, "The Value of a Causal Loop Diagram in Exploring the Complex Interplay of Factors that Influence Health Promotion in a Multisectoral Health System in Australia," *Health Research Policy and Systems*, vol. 16, no. 126 (2018), pp. 1-12.

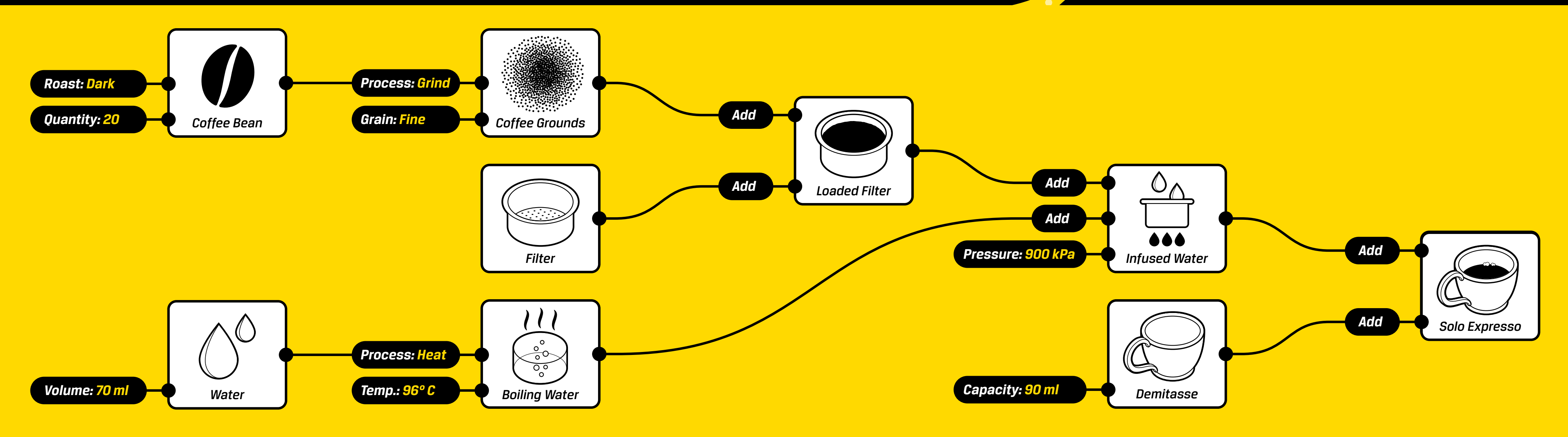
infographics



What methods work better? **Information graphics** sidestep many of the problems mentioned by distilling systems down to their most salient parts and illustrating whole-part relations both selectively and pictorially. Infographic designers rely on some shorthand techniques found in graphical notations, such as arrows and other basic symbols. Yet information graphics are composed flexibly with heightened awareness of the cognitive limitations of viewers. It is perhaps no coincidence that infographics grew in popularity just as the evolution of notations slowed. A core argument behind *Escalade* is there is great potential in evaluating infographics and applying lessons about what works best to notations. That starts by applying "show, don't tell" principles and using subject-specific images. Many infographic devices, such as call-out boxes, can also be used.

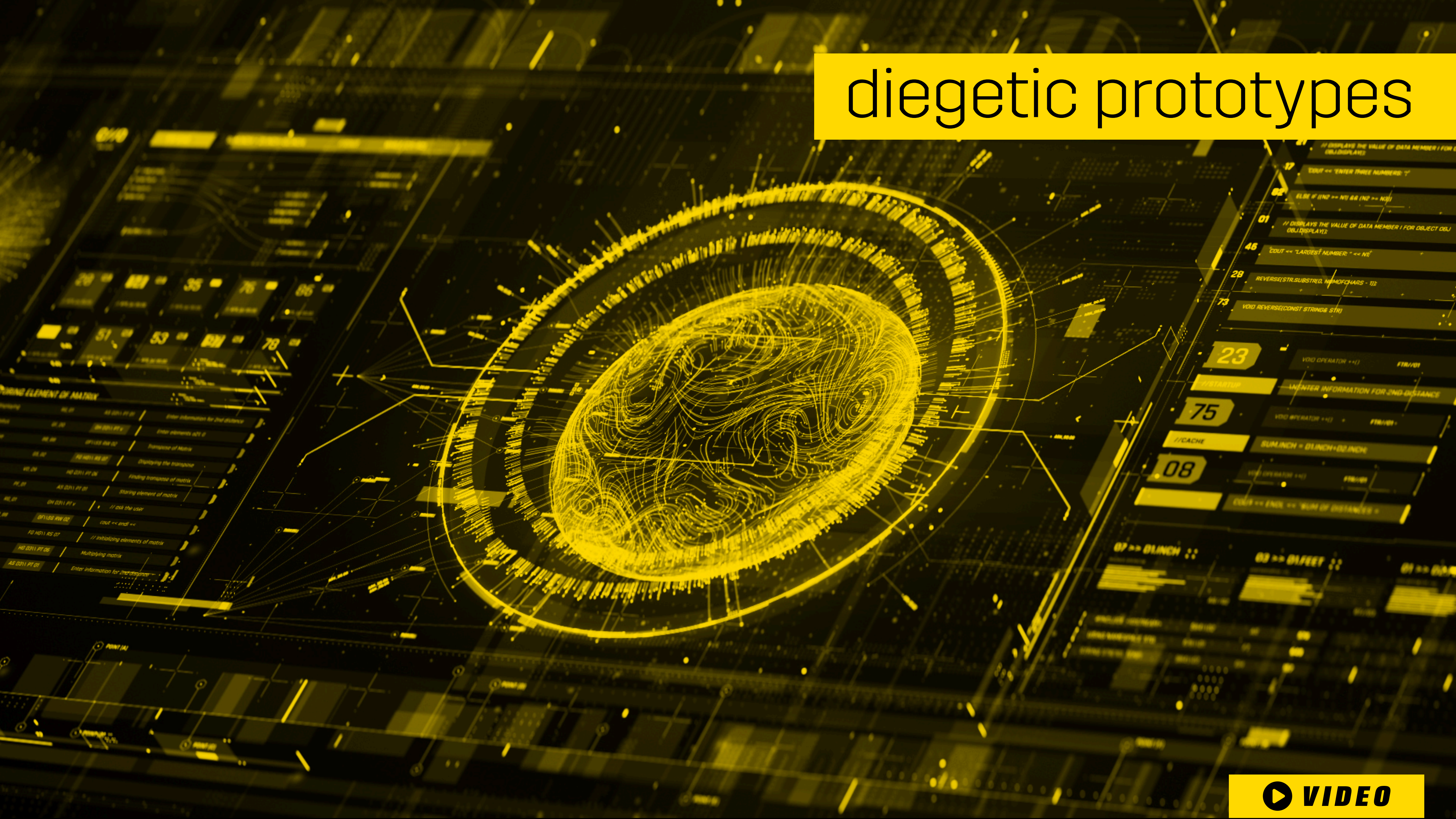
visual programming





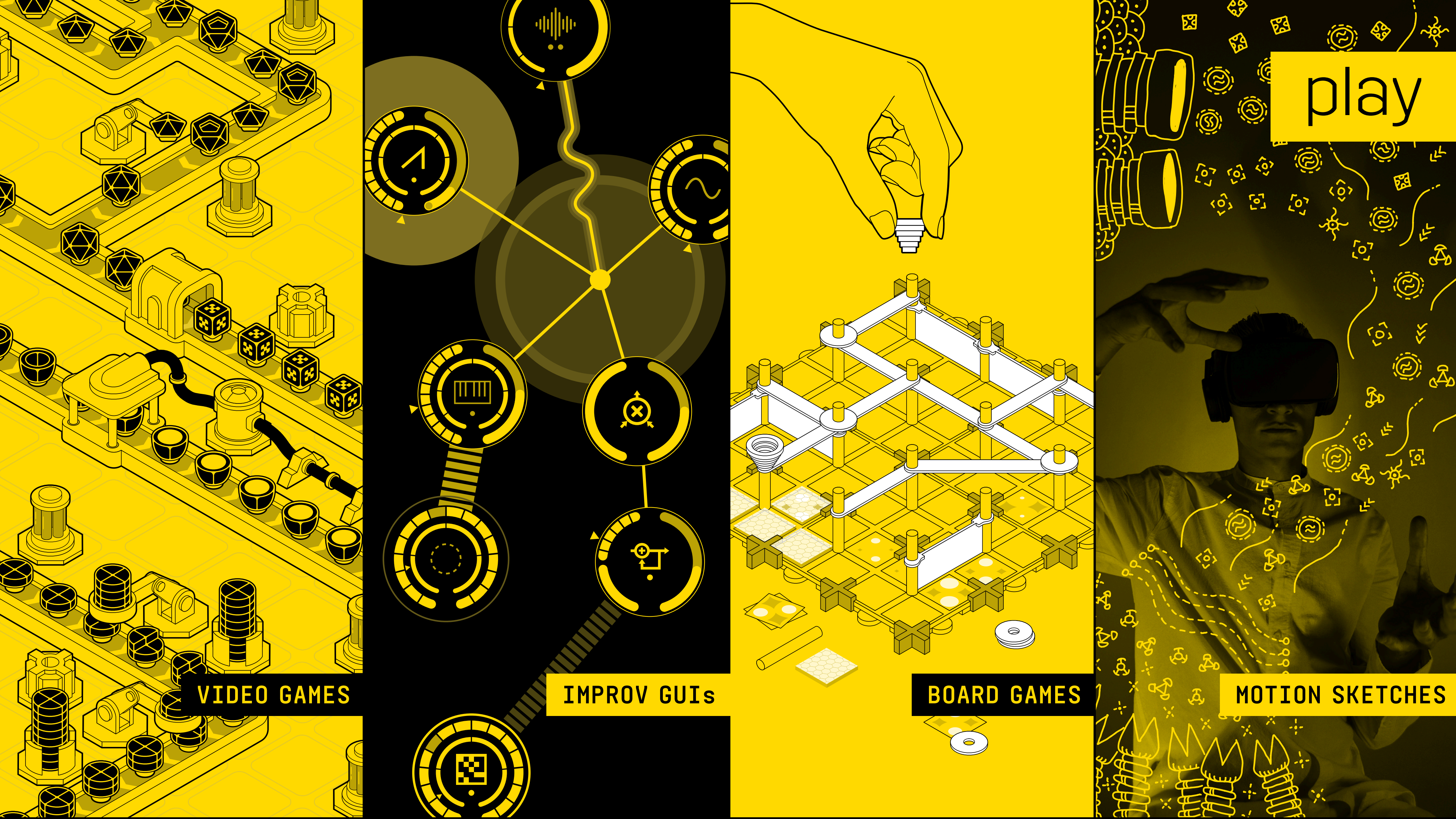
Another inspiration is **visual programming**. Not only do these interfaces display systems, they are functioning systems in their own right. These methods have a long history but are enjoying a renaissance thanks to technological advancements. The conventions remain in flux. Layouts struggle with some of the same entanglement problems plaguing system notations in general. Nonetheless, visual programming offers clues about how to model system operations visually and how to attach functional information to graphical nodes and links. As the bottom example shows, visual programming can be used to explain commonplace routines that are easy to relate to, breaking them down into discrete functions as a computer programmer would. As human-made systems rely more and more on software code, these methods become increasingly relevant.

diegetic prototypes



Fictional User Interfaces (FUIs) in futuristic films epitomize "eye candy" insofar as they awe us with technological patterns that are mostly meaningless. However, on closer inspection, these **diegetic prototypes*** offer insights into how to: present layers of system information; peer through boundaries to inspect underlying systems; present particle flows moving through systems; add secondary controllers and indicators onto interface widgets; overlay information onto spatial maps; the list goes on. Their influence on cultural expectations is hard to ignore. They contribute to a *visual vernacular* that has influenced the interface design of consumer gadgets, video games, and technical equipment, including medical-scanning technology.

* David A. Kirby, "The Future is Now: Diegetic Prototypes and the Role of Popular Films in Generating Real-World Technological Development," *Social Studies of Science*, vol. 40, no. 1 (2010), pp. 41-70.



Speaking of which, system visualization continues to evolve within interactive media. Several video-game genres revolve around system building. For example, two dozen factory-building games are slated for release this year. Touch-screen instruments for live performances allow musicians to arrange widgets into constellations of links. Each node and link is both a controller and status indicator. System-themed board games allow a group of players to map systems on a shared space.* Animated sketching and interactive art bring system dynamics alive. I love James Paterson's whimsical, animated ecosystems made with *Looom* and *NormanVR* (PressTube.com). In all cases, a sense of **playfulness** motivates viewers to explore the system.

* Board game shown is Co-gnito: Georgia Panagiotidou et al, "Co-gnito: a Participatory Physicalization Game for Urban Mental Mapping," *C&C '22: Creativity & Cognition*, June (2022), pp. 284-297.

motivational task loops

	PARSE	INTERPRET	REASON	APPLY
VIEWER	Attractive Aesthetics Clear Sight Lines Visual Hierarchies Intreguing Revelation Many Points-of-Interest Stimulating Variation Distractions Removed Good Spatial Separation Shape Contrast of Objects Comfortable Resolution	Pictorial Vividness Contextual Learning Aids Dual Coding (Text + Image) Meaningful Juxtapositions Thoughtful Distinctions Few Memorized Encodings Less Concept Vagueness Encoding Consistency No Multi-meaning Visuals[†] Textual Conciseness	Enriching Challenge System Mechanisms Clear Access to Elaboration Discovery Satisfactions Constructive Uncertainty Low Memory Demands No Baseless Causal Leaps Whole/Part Relations Clear	Situational Relevance Signs of Progress Learning from Setbacks Reference Functionality Reality Correspondence Multi-use Flexibility
MODELLER	(Re)Arrangement Ease Interface Fluidity Versioning Ease Autocomplete Aids Easy Scale Construction	No Missing Visual Objects[†] No Redundant Vis. Objects[†] Can Add Custom Visuals Handy Visual Libraries Viewer Limitations Clear	Brainstorming Ease Map As Extended Mind Accommodes Unknowns Ease of Scalar Switching Enriching Exploration	Clear Use Cases Known Context of Use

It is worth thinking more about **motivation** within the viewer task-loop given the complexity of many maps. What drives viewers to stay interested? What drives modellers to research and design elaborate maps? Apps that are "sticky" and video games with compelling "game-play" keep users captivated. Some techniques tap into human psychology in cynical and manipulative ways, as with the addictive *gamification* using token rewards and status goals—creating a *compulsion loop*. The satisfactions of genuine learning and discovery can be just as enticing if incremental gains in knowledge are insightful and challenging, plus the experience enables rather than annoys.*

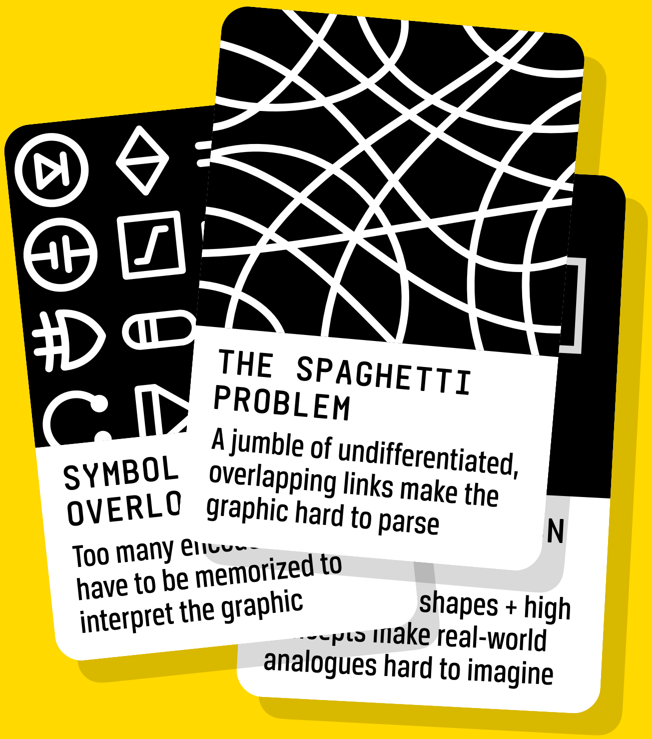
* The *Hest* visual programming language is being built with this idea at its heart, as discussed by its creator in: Ivan Reese, "Rabbit Hole," *Hest* [podcast], ep. 37, April 17 (2023).

† Daniel L. Moody, "The 'Physics' of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering," *IEEE Transactions of Software Engineering*, vol. 35, no. 6 (2009), pp. 756-779.

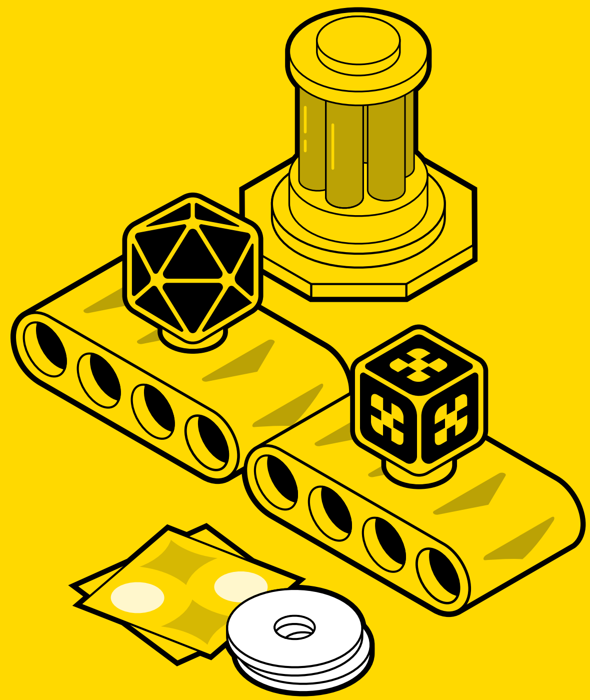
visual vocabulary

What does the larger literature say about systems? Do these notations leave out anything important? The ***Visual Vocabulary of Systems*** is a codex of the elements and dynamics of system with relevance across disciplines. System theories from the natural sciences, social sciences, design disciplines, and management disciplines were surveyed. All the elements and dynamics were grouped into six marquee categories: drivers, signals, states, boundaries, relations, and domains. The *Visual Vocabulary* has a unified visual syntax that illustrates each item, so all of these items can be depicted visually in the abstract. Existing notations show a strong bias in favor of only half of the categories: drivers, signals, and relations. Ultimately, the goal is to achieve better comprehensiveness. That may include adding the *Visual Vocabulary* to maps as an annotation layer.

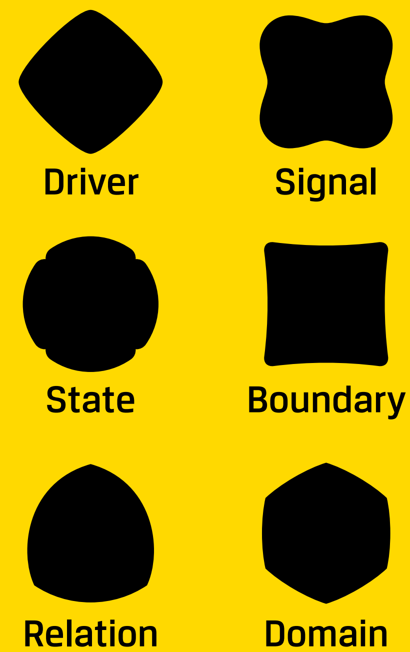
in sum



MALFUNCTIONS



INSPIRATIONS

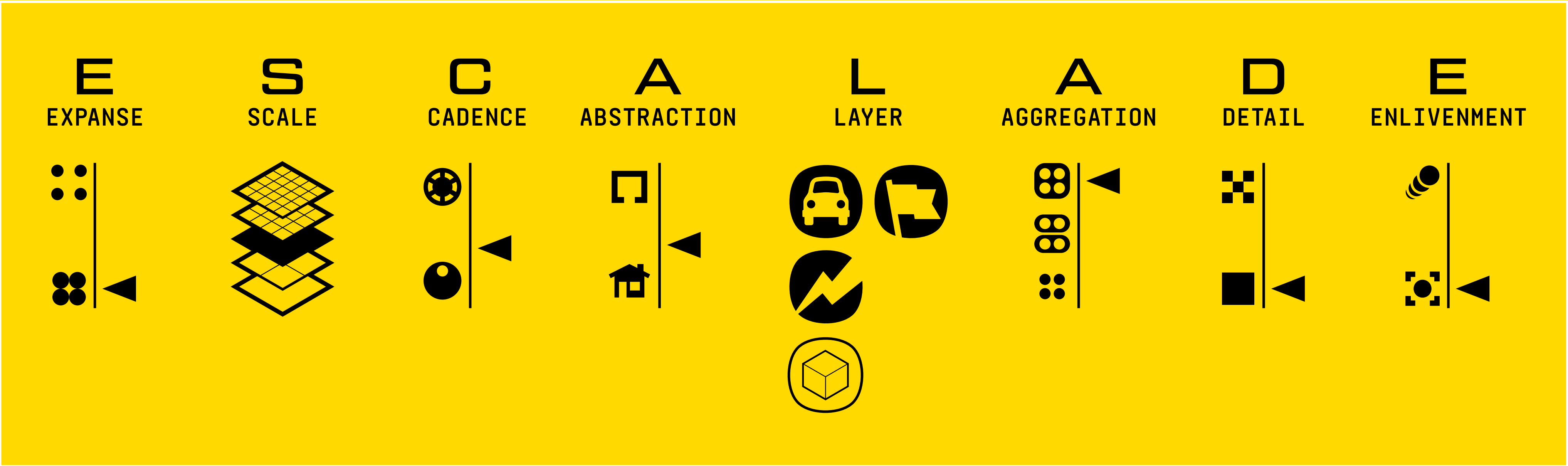


OMISSIONS

In sum, the **rationale** for a new information-design framework has three parts. First, commonplace graphical notations suffer from shortcomings that have stalled their development. Think of them as a card deck of problems that any notation should be able to overcome. Second, there is no shortage of inspirations for improvement but these opportunities remain under-explored because they require media cross-fertilization. Finally, there is now an inventory of the elements and dynamics of systems that can be used to assess completeness of any notation. Indeed, the *Visual Vocabulary of Systems* was originally designed to serve that role. The main weakness of graphical notations has been the makeshift ways the conventions were cobbled together in the first place. An information-design framework allows us to reappraise these conventions in a methodical way.

framework

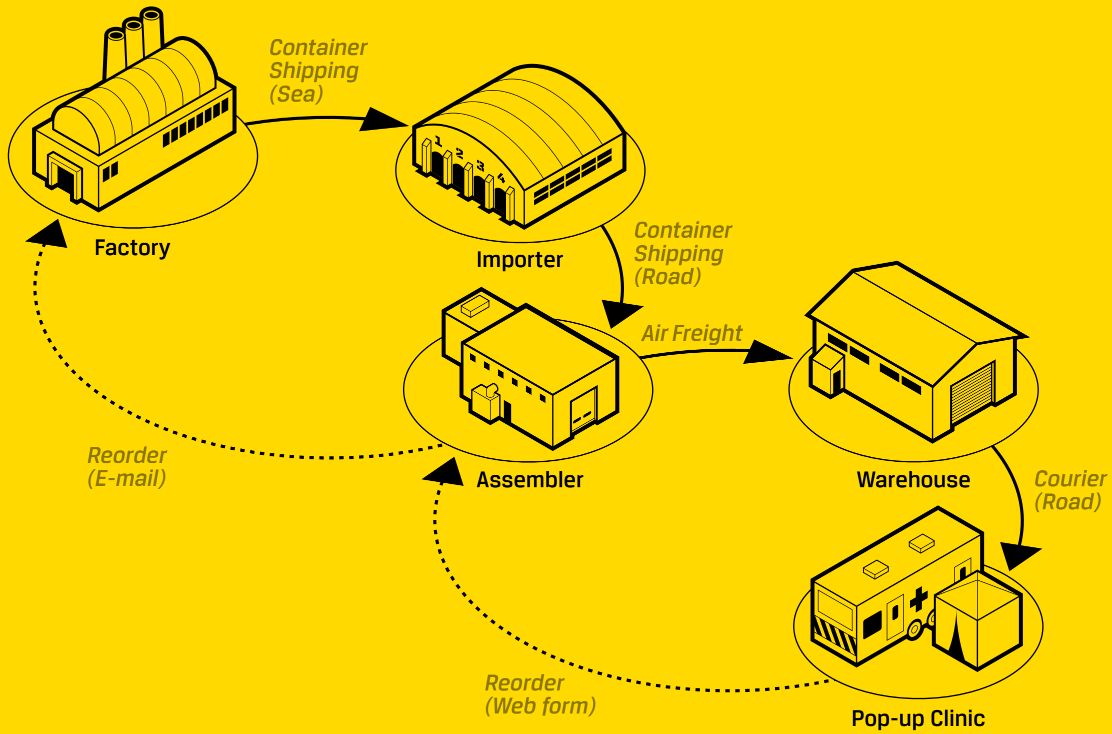
DASHBOARD

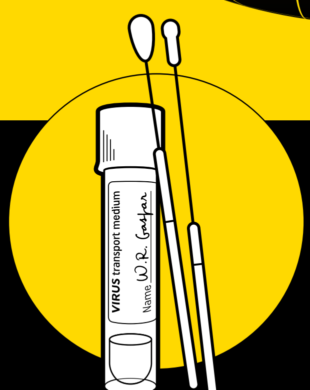


With those considerations in mind, let's return to the *Escalade* dashboard. When creating a system map, "dialling up" all dimensions to full realism would result in the map equalling the territory, a Borgesian absurdity that defeats the summarizing purpose of a map.* Conversely, "dialling down" all dimensions is equally unhelpful; a graphic too cryptic and nondescript to be recognizable as a particular system. Thus, the framework exists to help designers make intelligent trade-offs given the tensions inherent to mapping complex systems. The mark of a good **information-design framework** is how it organizes thinking around complicated subjects that are difficult to visualize with fidelity. Let's look at each dimension in detail.

* Literally so, as Borges parodied that scenario in the vignette, "On Exactitude in Science," in Jorge Luis Borges, *A Universal History of Infamy* (Harmondsworth: Penguin, 1975).

expanse



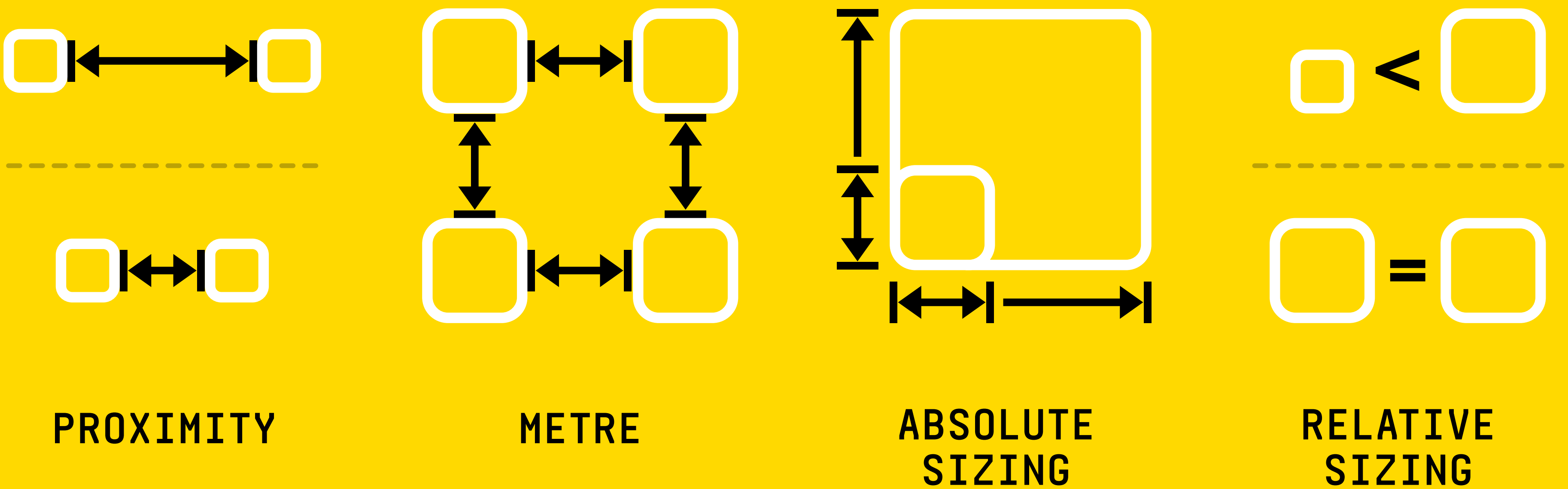


Viral-Infection Test Kit

The first item is **expanse**, or the treatment of space. In real life, system objects may vary in size and be scattered across vast distances or packed tightly into tiny spaces. Diagramming involves collapsing or expanding the spaces and adjusting sizes. The display medium imposes constraints and objects are arranged to make spatial associations that are meaningful for the viewer (*syntactics*).* Take the example of a test-kit supply chain, a network spanning the globe involving many infrastructural systems. Here we collapse it down into tidy circuits with meaningful juxtapositions.

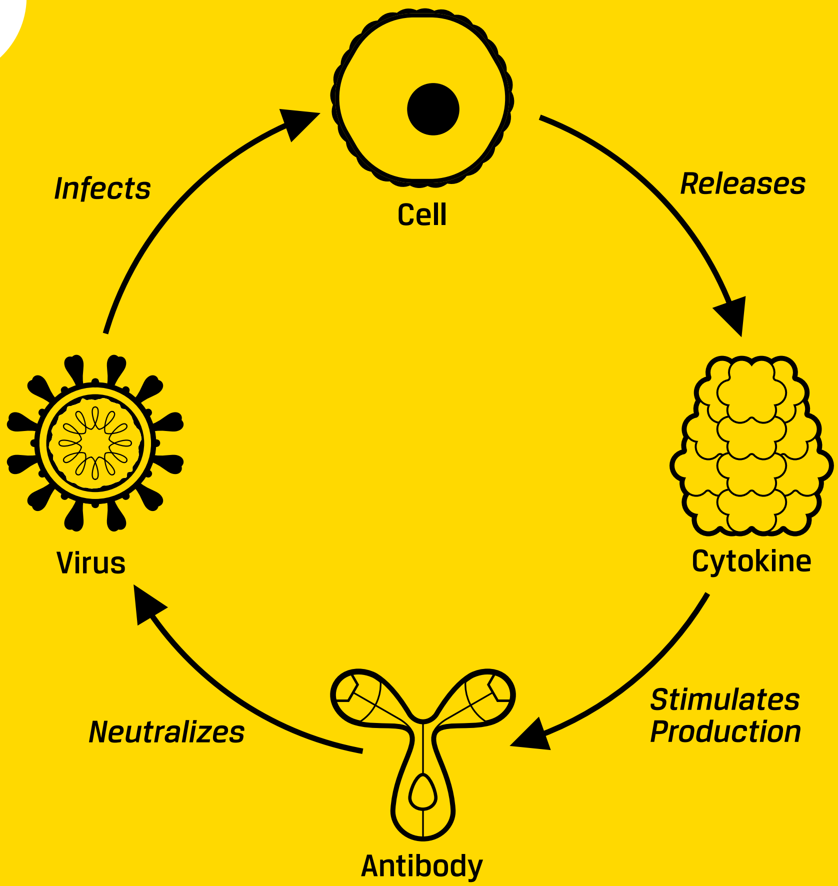
* Yuri Engelhardt, "Syntactic Structures in Graphics," *IMAGE: Zeitschrift für interdisziplinäre Bildwissenschaft*, vol. 5, no. 3 (2007), pp. 23-35.

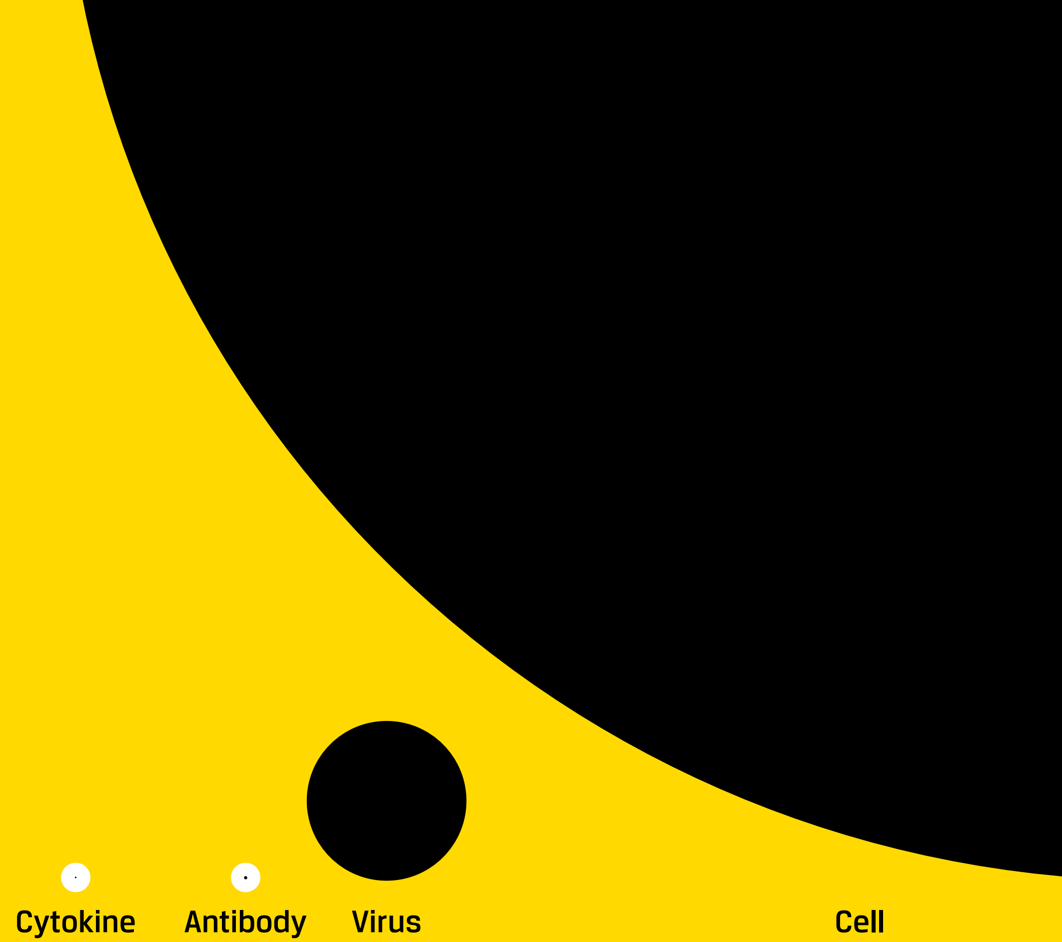
manipulations



Several **manipulations** are taking place, four of which deserve special attention. First, there is the treatment of *proximity* to fit the display and use spatial placement to better relate system elements to each other. Second, there is the *metre* of the objects, or the consistent treatment of in-between space to give object placement a spatial rhythm the eye can more easily parse. Third, there is the *absolute sizing* of objects so that they are readily identifiable without eye strain. Finally, the *relative sizing* is changed so that objects are a consistent size. Alternatively, sizes differences are adjusted to better reflect the role or relative importance of objects rather than their physical dimensions. All of these manipulations improve viewers' parsing and interpretation tasks at the cost of spatial-information loss, which can be reintroduced in other ways if relevant.

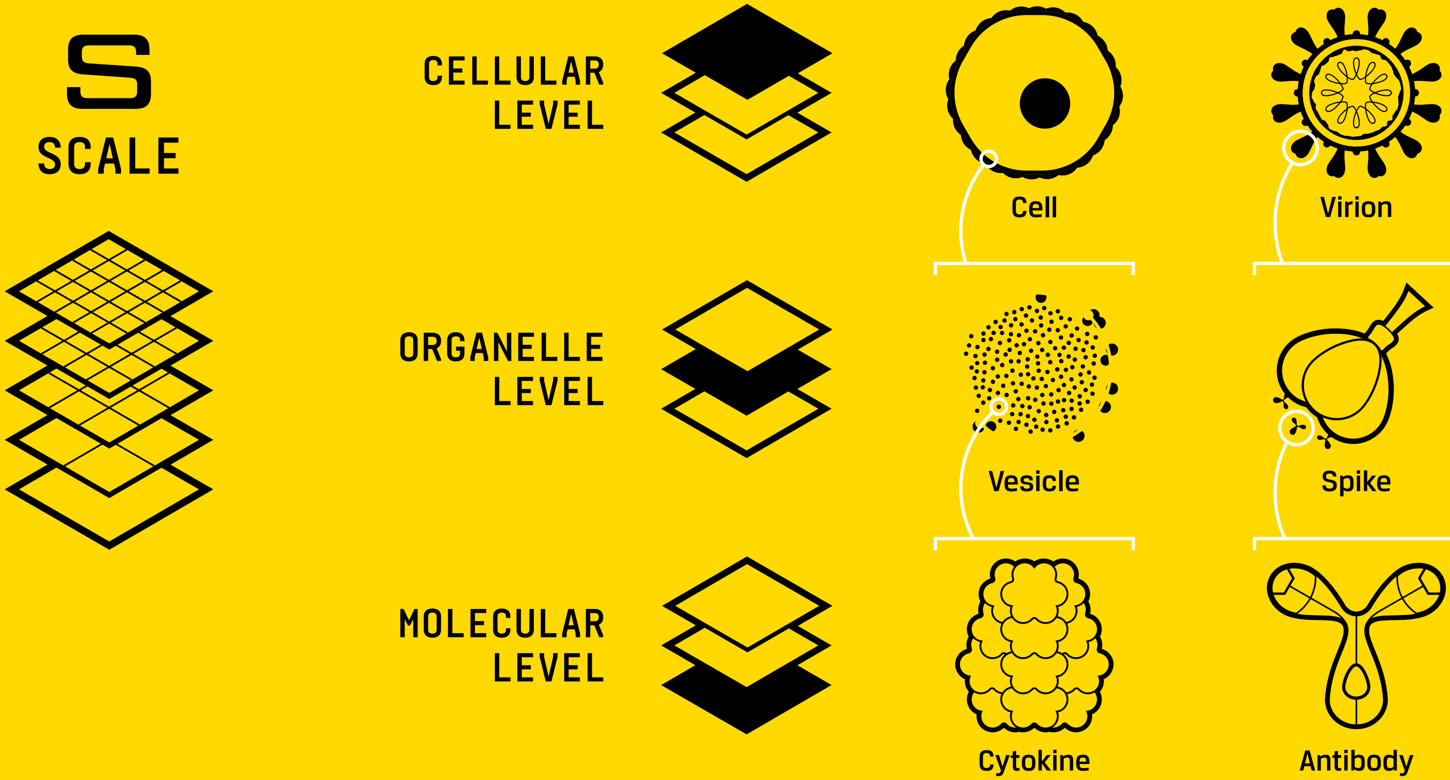
resizing





Consider an immune system example: the antibody-defence as an extremely simplified loop. Same-size objects are arranged into a causal loop. The consistent **sizing** of highly differentiated shapes eases viewing. Yet these objects are far from an equivalent size in real life. We can still imagine these objects interacting. For example, the antibodies are expected to gum up the spike proteins of the virus (the parts that stick out), which then prevents it from docking with the cell's receptors. Likewise, we can imagine the cell secreting small plumes of interferons to signal the production of antibodies elsewhere (in B-cells). However, none of these mechanisms are shown; the arrows and arrow labels are doing a lot of summarizing work here. To be more explicit about mechanisms, we need to grapple with *step changes in magnitude*, which is the next framework item.

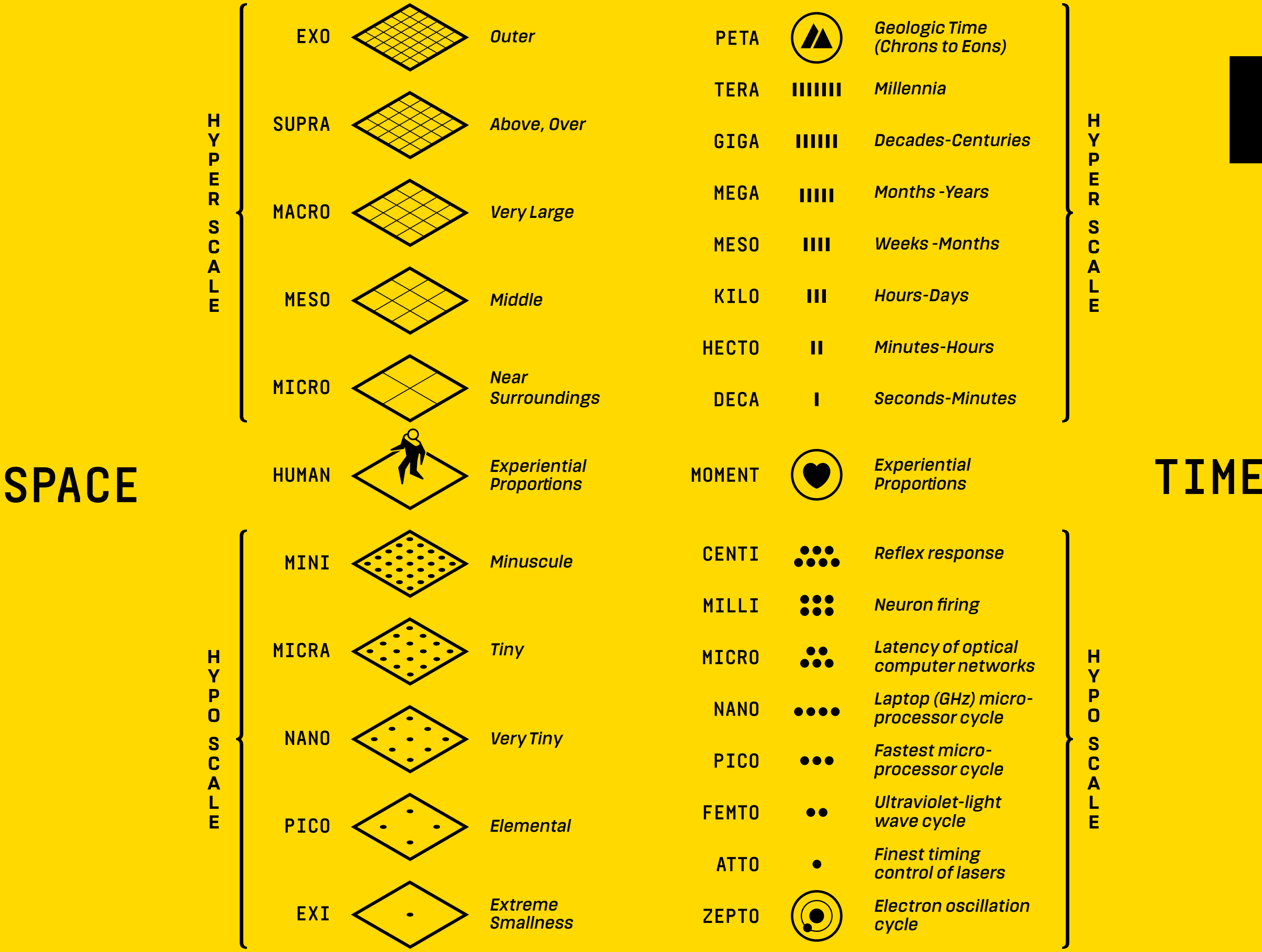
scales



The "S" in *Escalade* stands for **scale**. Scale is the level of analysis, or the objects and activities observable at a particular order of magnitude. In the example, the cytokines and antibodies are large molecules operating at the *molecular level*. The virus particle (virion) and cell operate on the same scale—the *cellular level*. If we want to show mechanisms, it might make sense to map activity at an intermediate level, that of the *organelle*.

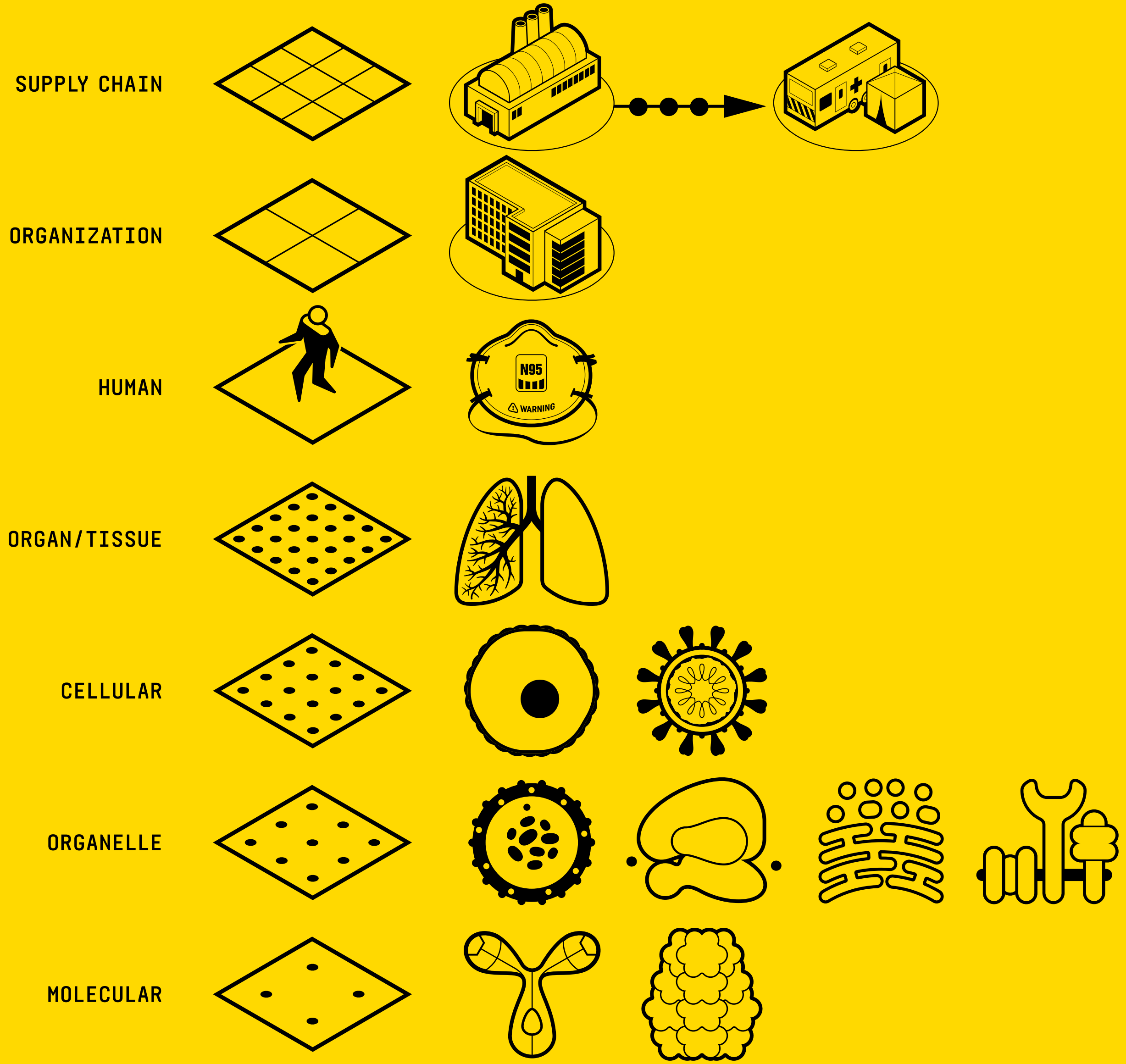
The *Escalade* dashboard depicts levels as stacked spatial planes (◇). System dynamics can be shown at the same level. Or if small-scale dynamics trigger changes directly at a higher level (what physics modellers call *subgrid-scale phenemena*) those interactions are made clear. That is made easier by using multiple scales rather than left to *fudge factors* (contrived adjustments) or mysterious forces emerging out of nowhere.

scales



Scale applies to both space and time. These generic scales provide a starting point for thinking about systems in a **multi-scalar way**. We normally think in experiential proportions: our immediate surroundings at the human scale for a moment of time (a few seconds). Most systems would escape our attention if we only ever noticed things at that scale. Some systems sprawl across vast distances that cannot be seen all at once at the human scale. They may change so slowly that they seem static. Conversely, systems may work at scales too small for the naked eye to see. Even if viewable, their speed make them appear as a blur. Thankfully, we have instruments that allow us to observe and gather data about these scales. The challenge for diagramming is that most elaborate systems operate on multiple levels at the same time.

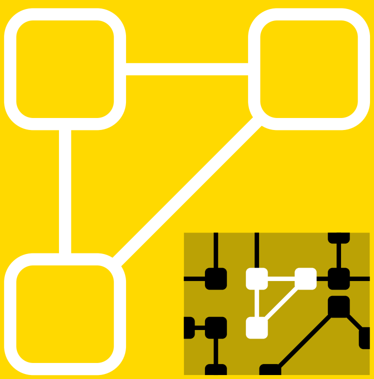
scales



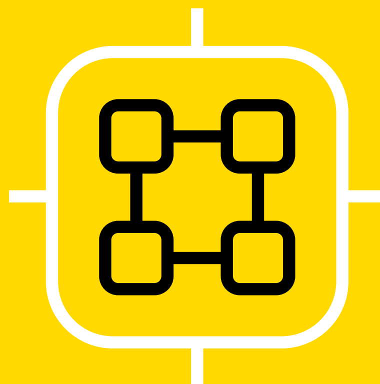
We can order objects in our health-system examples into a table to see how many levels of scale are involved. These step differences have practical implications for how things work. Test kits are used to measure antibodies (*serology tests*), proteins on the surface of the virus (*antigen tests*), or nucleic acids that make up virus proteins (*molecular tests*). Thus, kits are human-scale instruments that measure traces of extremely tiny objects.

Some tests can be self-administered but most require processing in a laboratory. Elaborate logistical systems gather samples, process them, and report findings to the relevant actors. The development and manufacturing of the kits are even more elaborate systems that potentially span the globe. It is remarkable how existing graphical notations either obscure these scale differences or show only one level of scale at a time.

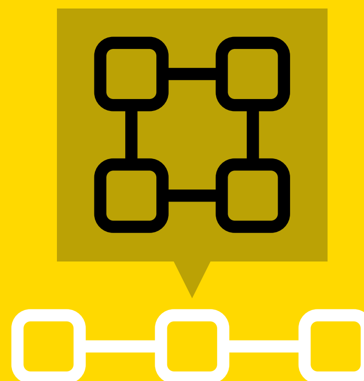
graphical devices



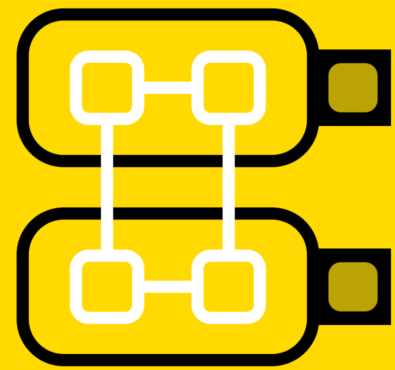
HUD MAP



NESTING



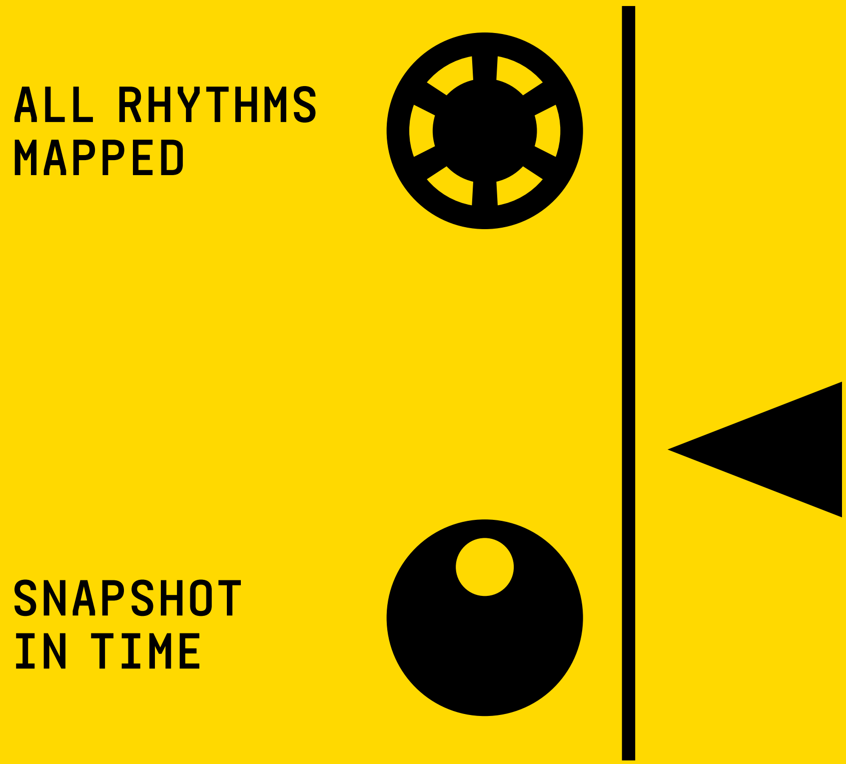
CALL-OUTS



BOUNDARIES

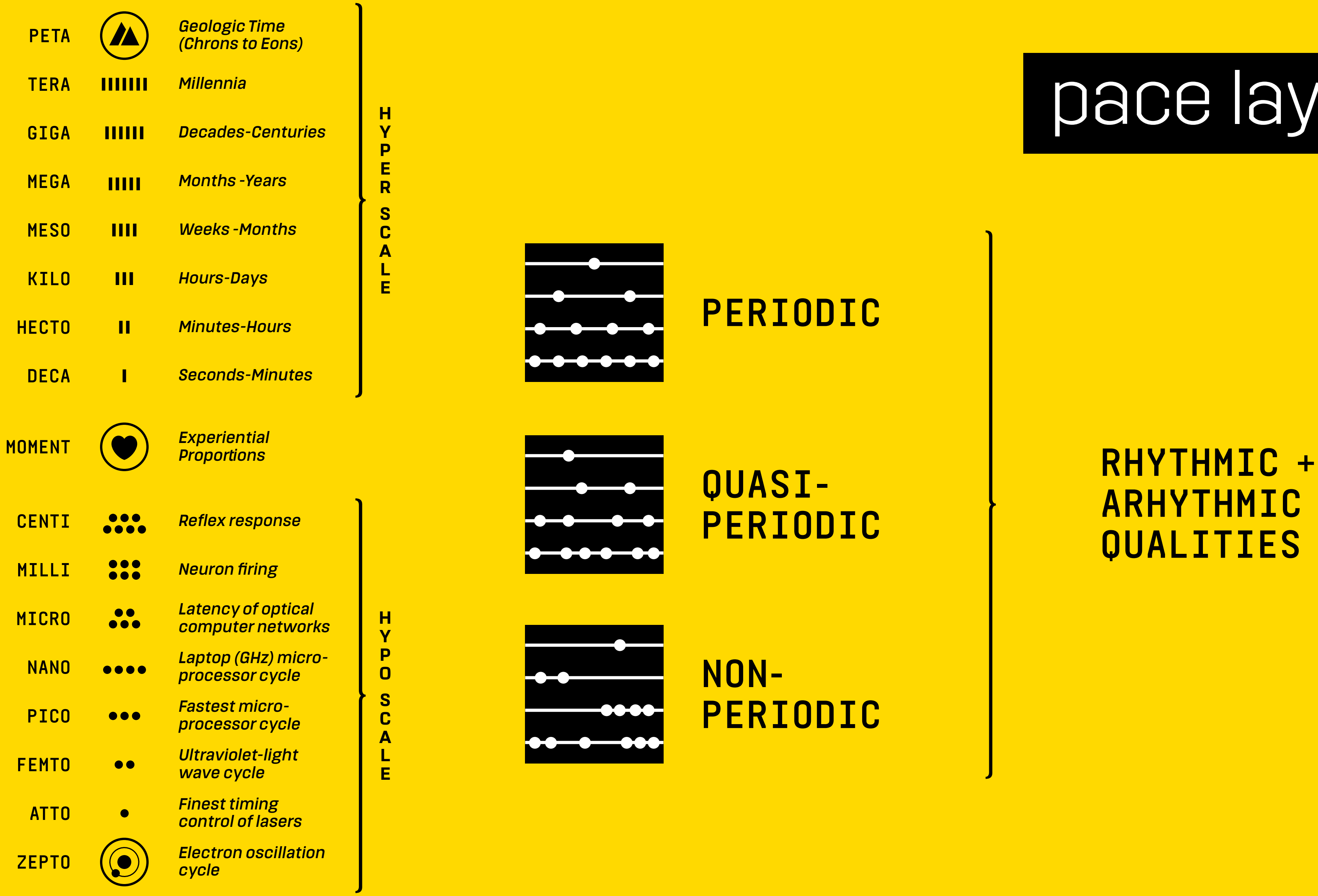
There are many **graphical devices** for showing different levels of scale. Here are a few examples. A *HUD* (Heads-Up Display) shows positioning at a higher level within a small, superimposed window. Systems and routines can be *nested* within nodes. Some screen interfaces allow the viewer to zoom into (or out of) objects or spaces to observe lower (or higher) levels. Alternatively, separate windows can show zoomed-in and zoomed-out views. Call-out boxes point to objects and elaborate on system activities at another level. Systems can be shown spanning boundaries, with an encompassing boundary representing a node at a higher level (with the node object attached to the boundary). As mentioned, it is also common in computer science to portray levels as *stacks* (not shown), with higher- and lower levels of analysis ordered vertically.

cadence



What about time scales? The "C" in *Escalade* means **cadence**, or the encoding of a system's temporal rhythms into the map. A *synchronic* map presents a snapshot at a single point in time. Conversely, with a fully *diachronic* map, all cycles, time signatures, and state changes are described. In between those poles, a selection of cyclical rhythms and state changes are included. The time dimension poses challenges to mapping. First, maps are arranged spatially, which lends itself to certain time markers but can send mixed signals if temporal graphics rely on spatial metaphors (as timelines do). Second, precise times may not be attributable to system processes, particularly if processes with different rhythms are aggregated together. Ranges may have to suffice. Third, not everything will have a regular rhythm, so a few additional distinctions should be kept in mind …

pace layers

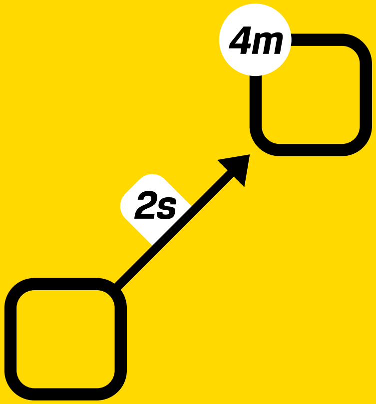


Revisiting the topic of scales, there is a correlation between levels of analysis and the frequency of change. Stewart Brand coined the term **pace layer*** to stress that small-scale systems move very quickly (such as ecosystems full of microbes), while large-scale ones move very slowly (such as systems of geological flows underneath the ground). Some change happens with great regularity, or what can be called *periodic* change.

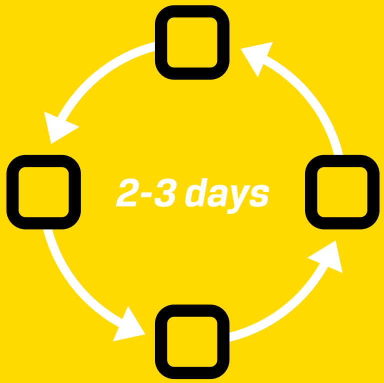
Others have a loose timing but nonetheless show a pattern. Both periodic- and quasi-periodic change count as rhythms, which can be depicted as part of a map to give the viewer a sense of the frequency of interactions. Even irregular happenings can be noteworthy findings to be reported in the map as probabilities.

* Stewart Brand, *The Clock of the Long Now: Time and Responsibility* (New York, NY: Basic Books, 1999), pp. 33-40. The general idea originated with architect Frank Duffy, who called them "shearing layers."

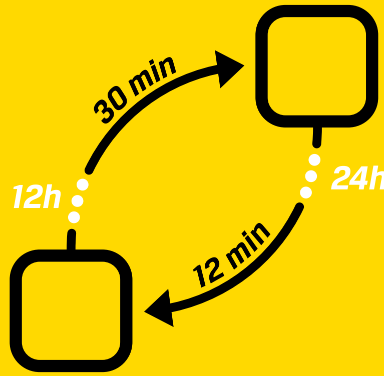
graphical devices



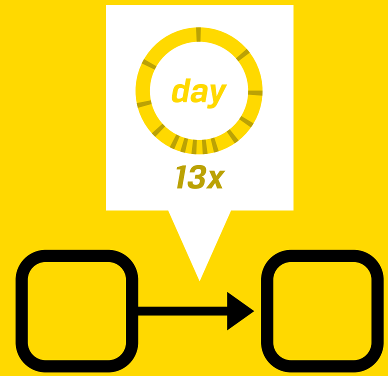
MARKERS



STRINGS



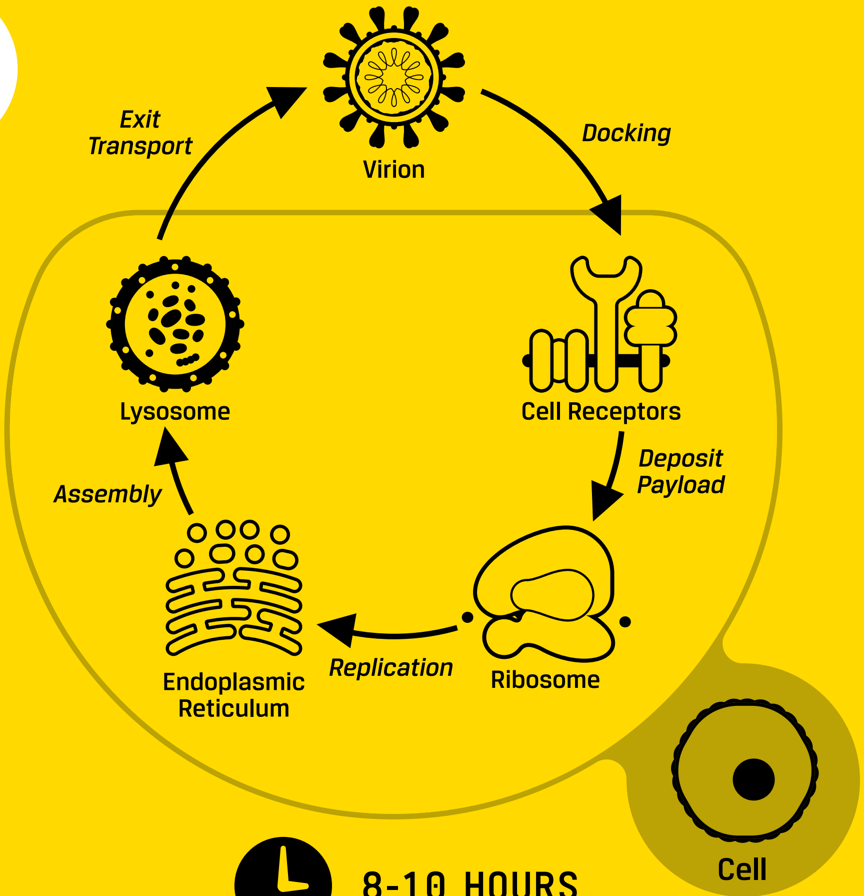
LAGS

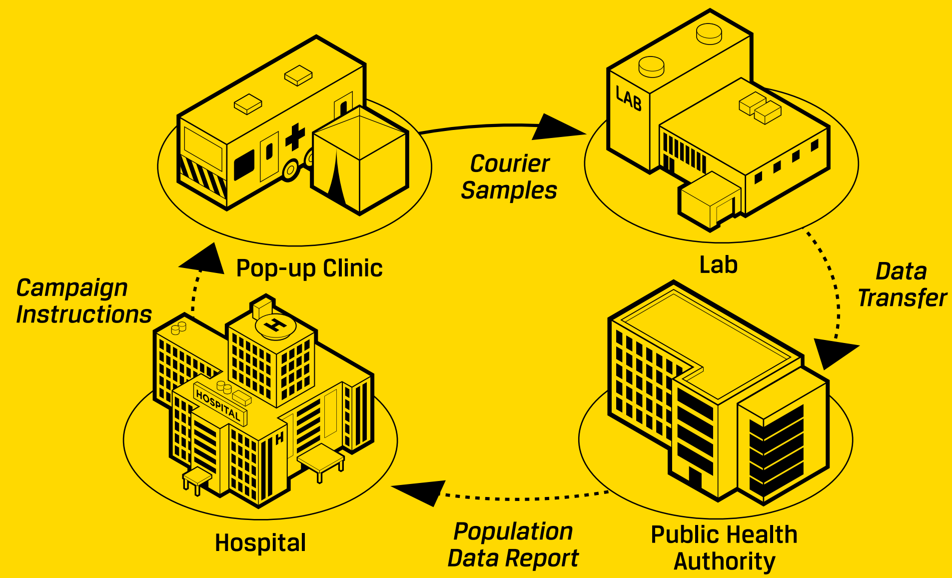


SCHEDULE

How do we add time markers to system map? Duration labels can be added (as values or ranges) to any object that represents a happening or process, or a non-happening (wait, lag, or period of dormancy). Multiple node-and-link strings can be assigned a duration, although with existing notations that is easiest with isolated loops. Interruptions to a link may indicate delays or lags, which can be tagged with times. A scheduling call-out or annotation may show a more sophisticated widget showing rhythmic patterns. As mentioned, some sort of *modal separation* should exist to avoid timing spatial metaphors interfering with spatial mapping. In that spirit, some opt to present system rhythms on a separate table that references processes on the map using cross-references. As a general rule, however, it is better to keep relevant information handy within a single map.

rhythms





8-10 HOURS

10-100 VIRIONS PER CYCLE

12-72 HOURS

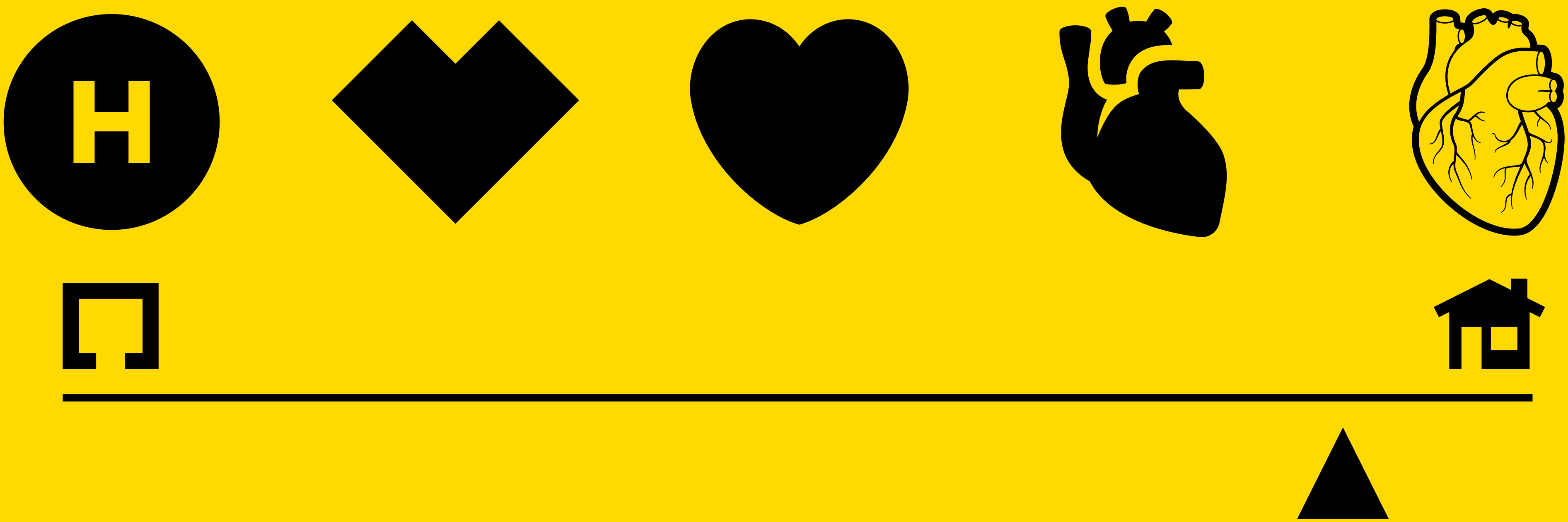
UP TO MAX THROUGH-PUT

Public health authorities aspire to operate surveillance systems that outpace the spread of viruses: data that moves faster than disease, so to speak. A virus-replication cycle ranges from a few hours to a few days. Covid-19 takes 8-10 hours to replicate in a human cell. Disease symptoms (*innate immune response*) take 2-3 days to emerge. Some symptoms (fever) slow virus replication, while others (coughing) increase spread between people.

Antibody production (*adaptive immune response*) may take 6-8 days to achieve effectiveness. Developing a mass-population testing system for a novel virus is a major logistical feat. A labor-intensive lab-based arrangement will take at least 12 hours with high demand. By **showing time ranges** (and trend lines), it is not hard to see why elaborate testing systems will inevitably fail to keep up during a pandemic of a highly infectious virus.

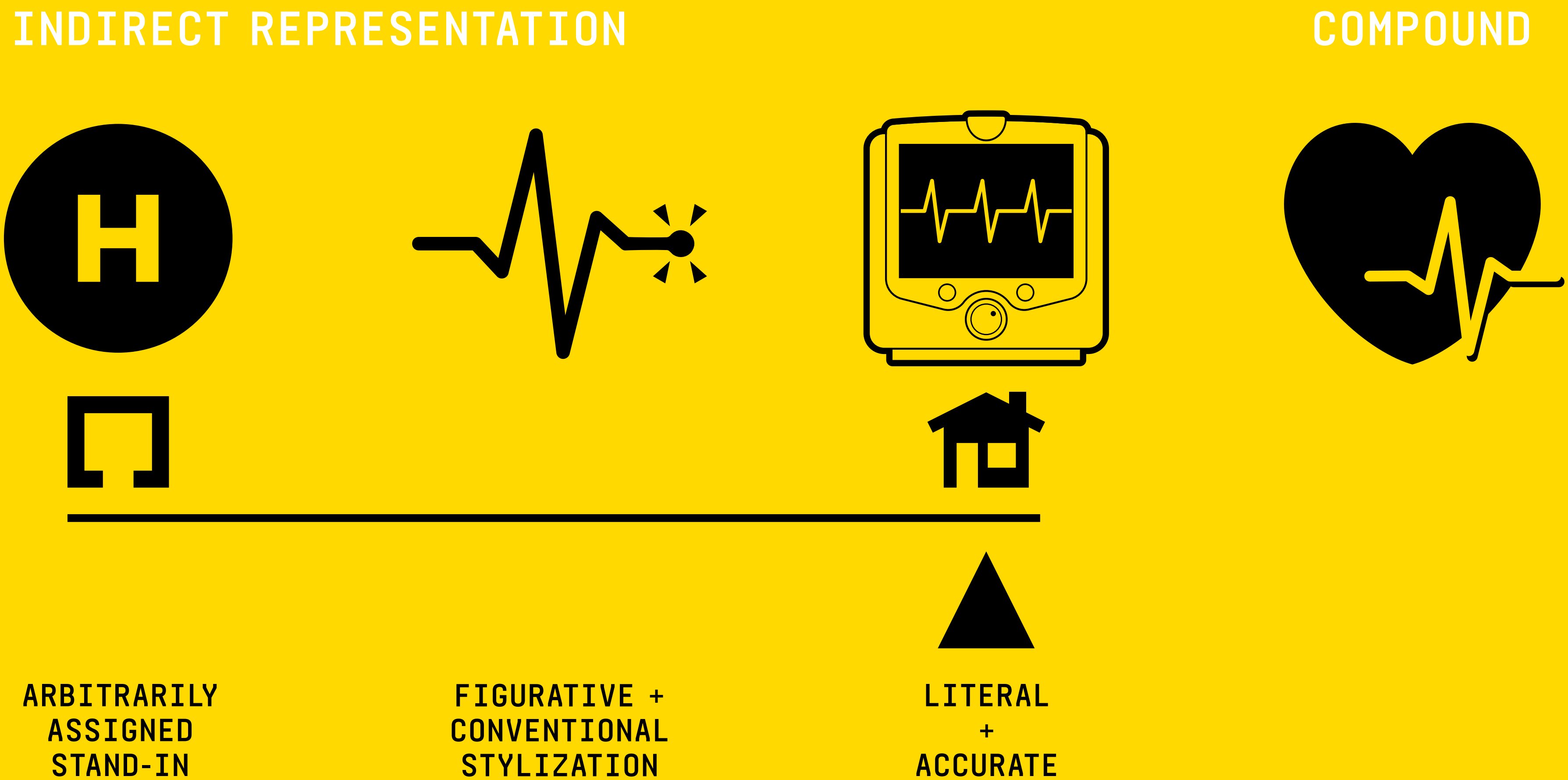
abstraction

DIRECT REPRESENTATION



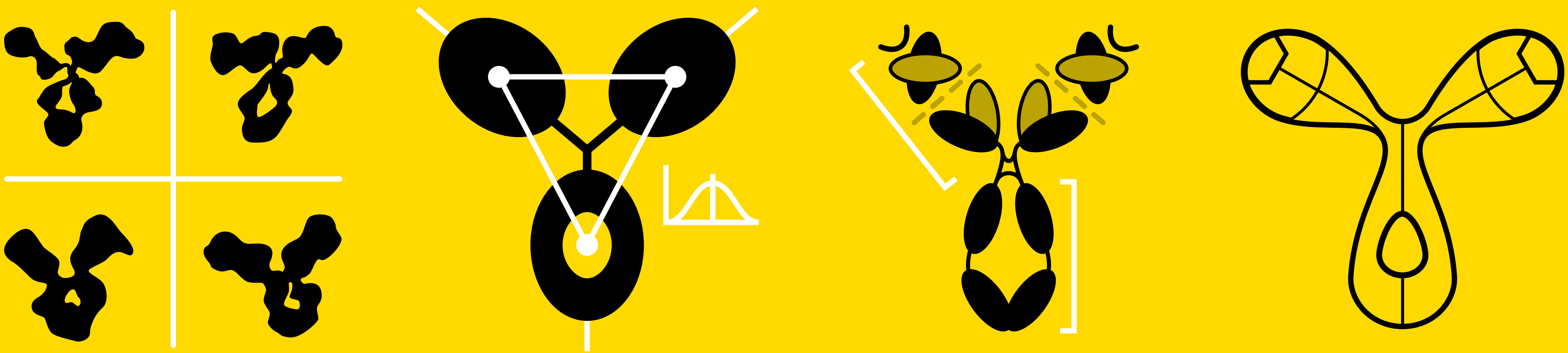
The first "A" stands for symbolic **abstraction**, or the visual accuracy of how objects are represented. Highly abstract stand-ins on a map have no visual correspondence to the actual thing as it exists in the real world. They merely act as arbitrary symbolic placeholders. Any shape or generic symbol is assigned a meaning that viewers are expected to learn. Less abstract images have a literal correspondence to real-world objects and are depicted accurately. Many of us will have no first-hand experience with the parts of most systems. Think about a commonplace system, such as a car engine. What does a carburetor look like? Or a camshaft? Do all makes and models look alike? Is there an iconic form? Thus, literalism and accuracy do not guarantee recognizability. Stylization and cultural referents may have to clue viewers into what a system part is and how it works.

abstraction



A system object may not be directly representable. Perhaps we do not know what it really looks like. Or it does not have a discrete, stand-alone form that can be physically disentangled from other parts. Or its form is sprawling and diffuse, which does not lend itself to a space-efficient illustration. Or it is a verbal-linguistic concept with no material existence. In any case, we can represent those parts of the system indirectly. They may be associated with related objects, such as a heartbeat represented by the fluctuating line on a heart-monitor screen. That analogy may gain currency within the culture, making it a recognizable *trope* (non-literal signifier). Combining direct and indirect representations can also reinforce an intended meaning. Indeed, the compound image of figurative heart and fluctuating line is the logo of the VisionPlus conference.

ANTIBODY



REALISM (ACCURACY + VARIATION)

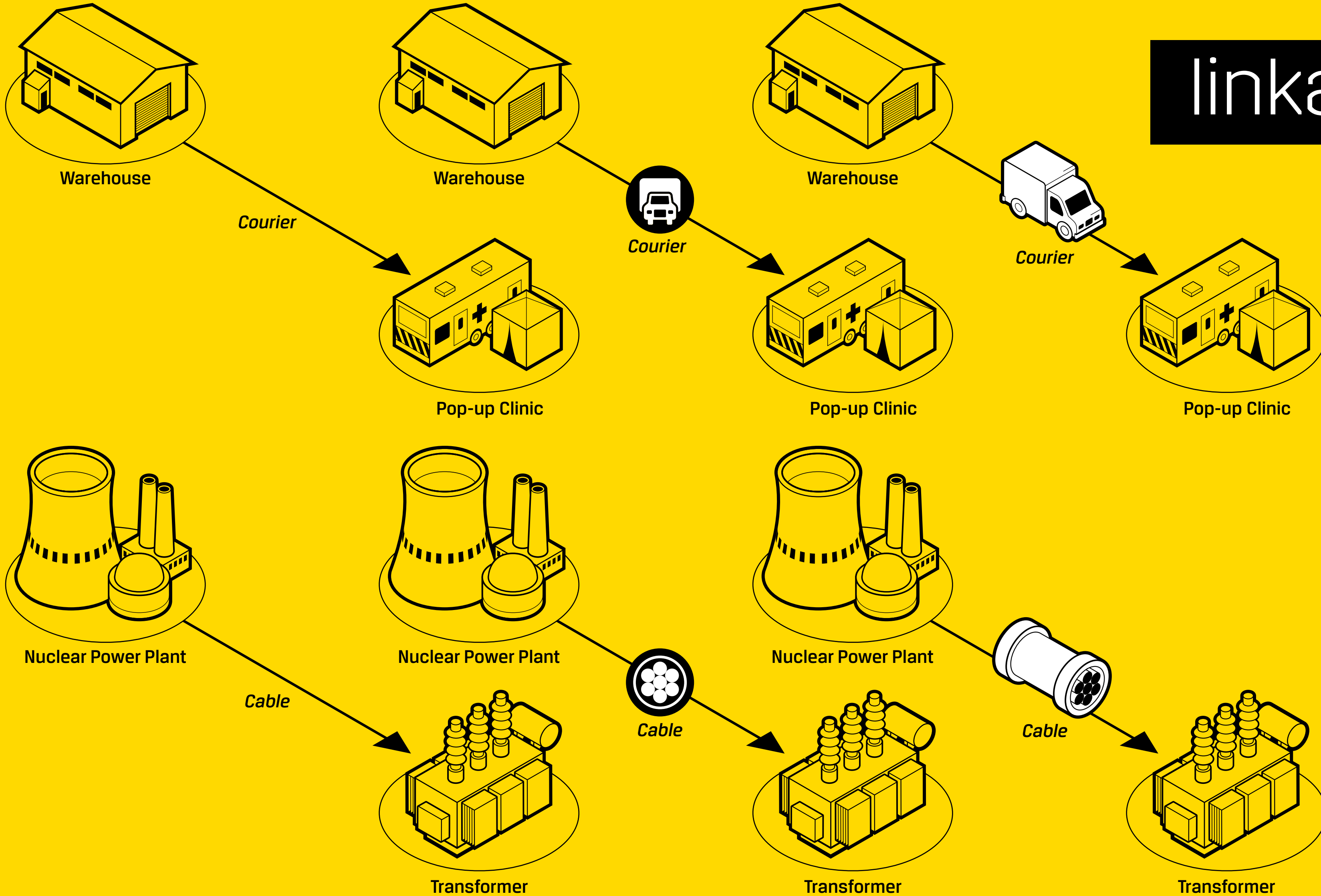
REPRESENT-ATIVENESS (TENDENCIES)

EXPRESSING FUNCTIONAL STRUCTURE

STYLIZATION (EMBELLISHMENT + CONSTRAINT)

Let's return to the antibody example. In reality, an antibody is a gnarled protein molecule with a flexible tripod shape. As with most objects, there is a lot of shape variability. A highly realistic depiction would be too quirky and amorphous to be easily identifiable. Statistical tendencies suggest an average shape to act as a representative stand-in. The underlying anatomy may indicate functional structures that imply how the object works or acts. Some stylization is applied to give coherence to the icons in the diagram as a set. A *ligne claire* style common to technical illustration is applied in this case. It entails some embellishment and constraint based on the style's aesthetic conventions. The overall goal is to create an **icon** that resembles the real thing despite variation across subtypes and cases. The risk is creating a hyper-real *simulacrum* that idealizes the object deceptively.

linkages

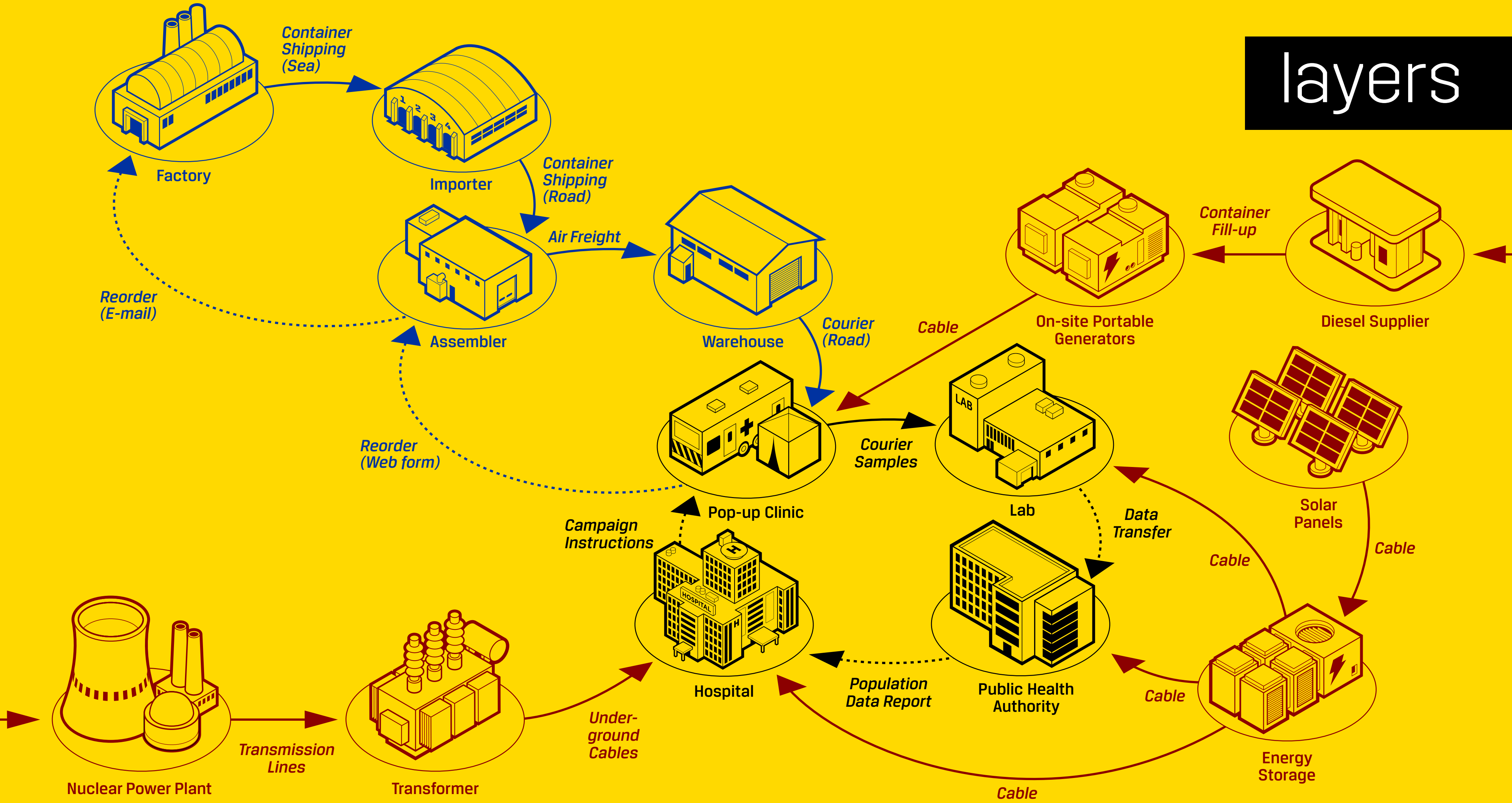


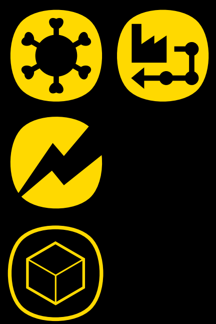
Adding images to the map is not limited to nodes, although few current notations work that way. Indeed, the overall map can be more "glanceable" if the **linkages** between pieces are obvious, even if it is just a reference to the medium or interface that mediates the relation. For example, in a supply chain, the means by which materials and signals are transported can be shown. Likewise for energy systems, the type of electrical conduit can be shown. Some verbs are complicated actions, happenings, or relations best represented by *transit nodes*; that is, minor nodes along links between major nodes. Differentiating transit nodes using size, shape, icon style, or other device makes them easier to parse. As mentioned, a chronic shortcoming of maps is the underspecification of links, making it difficult to imagine what is happening. Relying less on abstract concepts helps.

layers

The "L" in *Escalade* stands for **layers**. Natural- and human-made systems are deeply entangled with each other. For example, systems of public infrastructure are reliant on complex information-technology systems, which in turn are reliant on elaborate systems of electricity generation. Systems of food production are reliant on water-cycle ecosystems and waterworks for irrigation, which in turn are influenced by ecosystems that maintain regional climatic conditions. In a sense, every system is an *open system* that is somehow influenced by other systems. One goal of mapping systems is to disentangle these systems to understand how each works separately. Another goal is to selectively draw crosswalks between systems to understand how they influence each other. Thus, each system (or family of system) is presented as its own layer on the map.

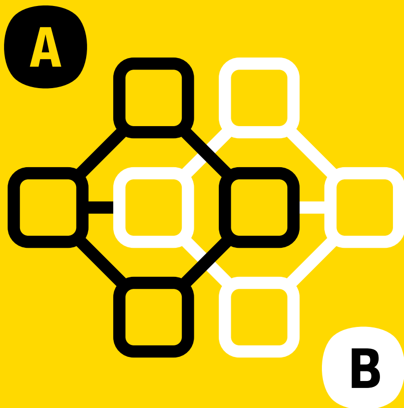
layers



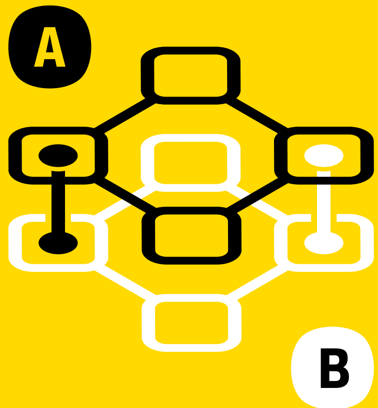


In the Escalade dashboard, solid icons inside super-ellipses (●) represent the systems being shown (left). In the example above, separate icons represent virus testing, supply-chain systems, and electrical systems. How are these systems portrayed? What are the main points of reference? In this case, physical structures are shown in isometric (distance-3D) perspective, represented by a cube drawn as a line icon within a hollow super-ellipse (O). Other referents could have been chosen, such as actor roles. We could expand the energy systems to encompass the supply chain too. And we could apply material supply chains to the energy systems. Among other things, that could be used to calculate the full energy- and material "footprints" of the system.

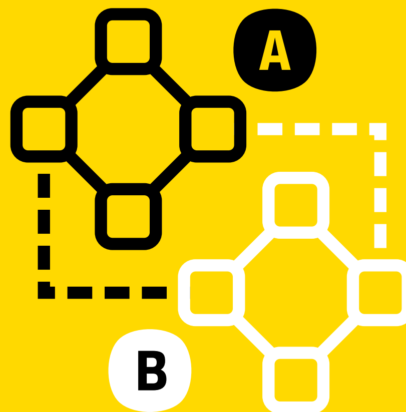
graphical devices



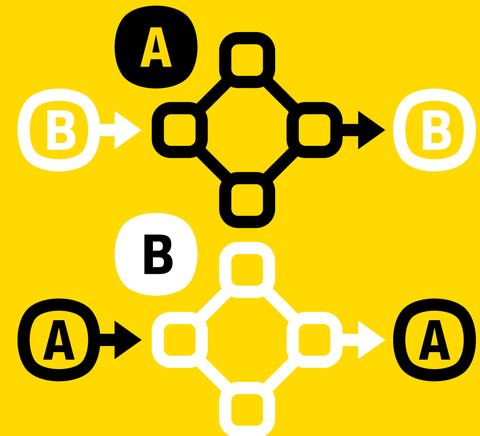
INTEGRATE (+ DIFFER-ENTIATE)



OVERLAY



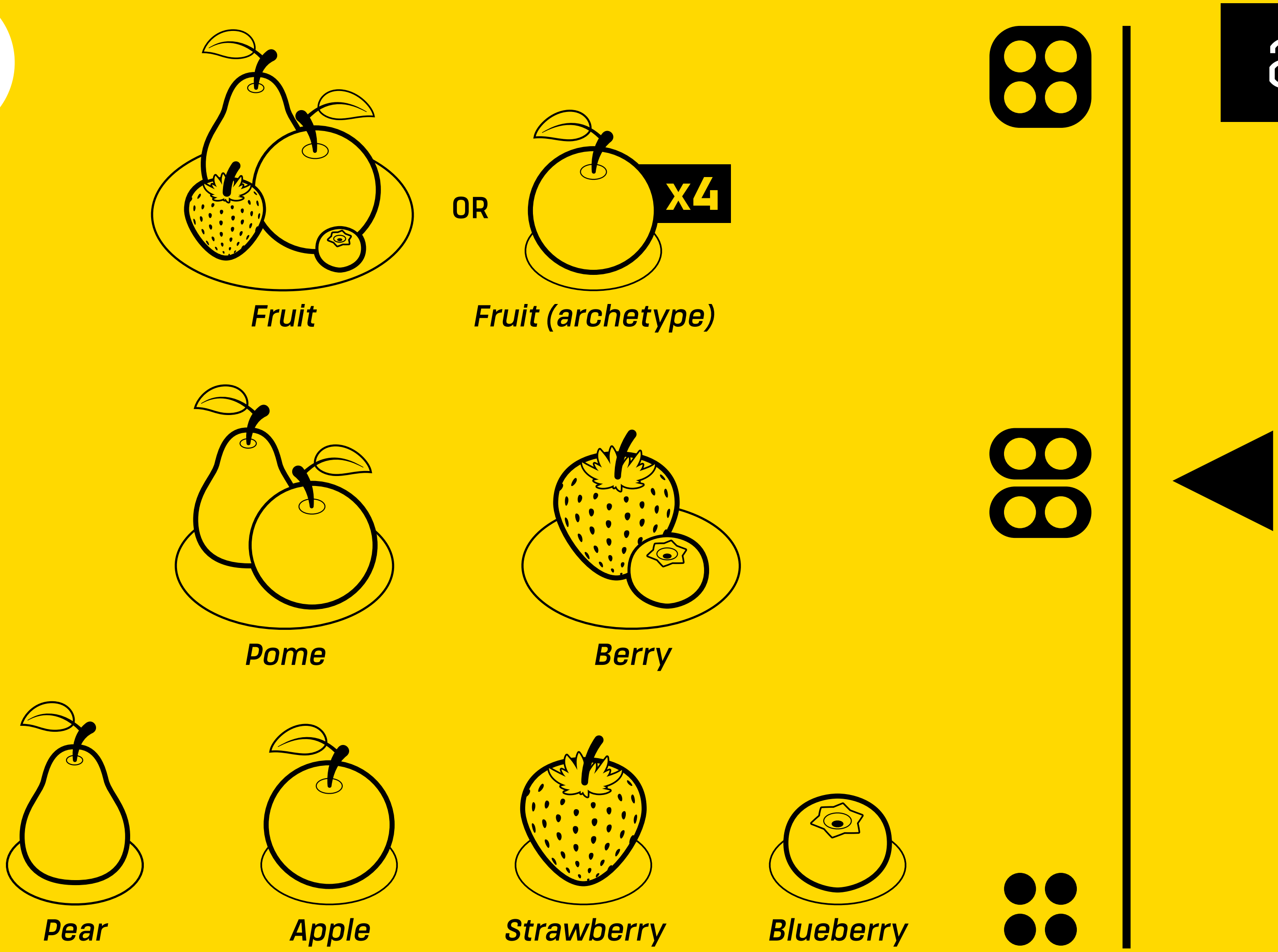
CROSSWALKS



JUMPS

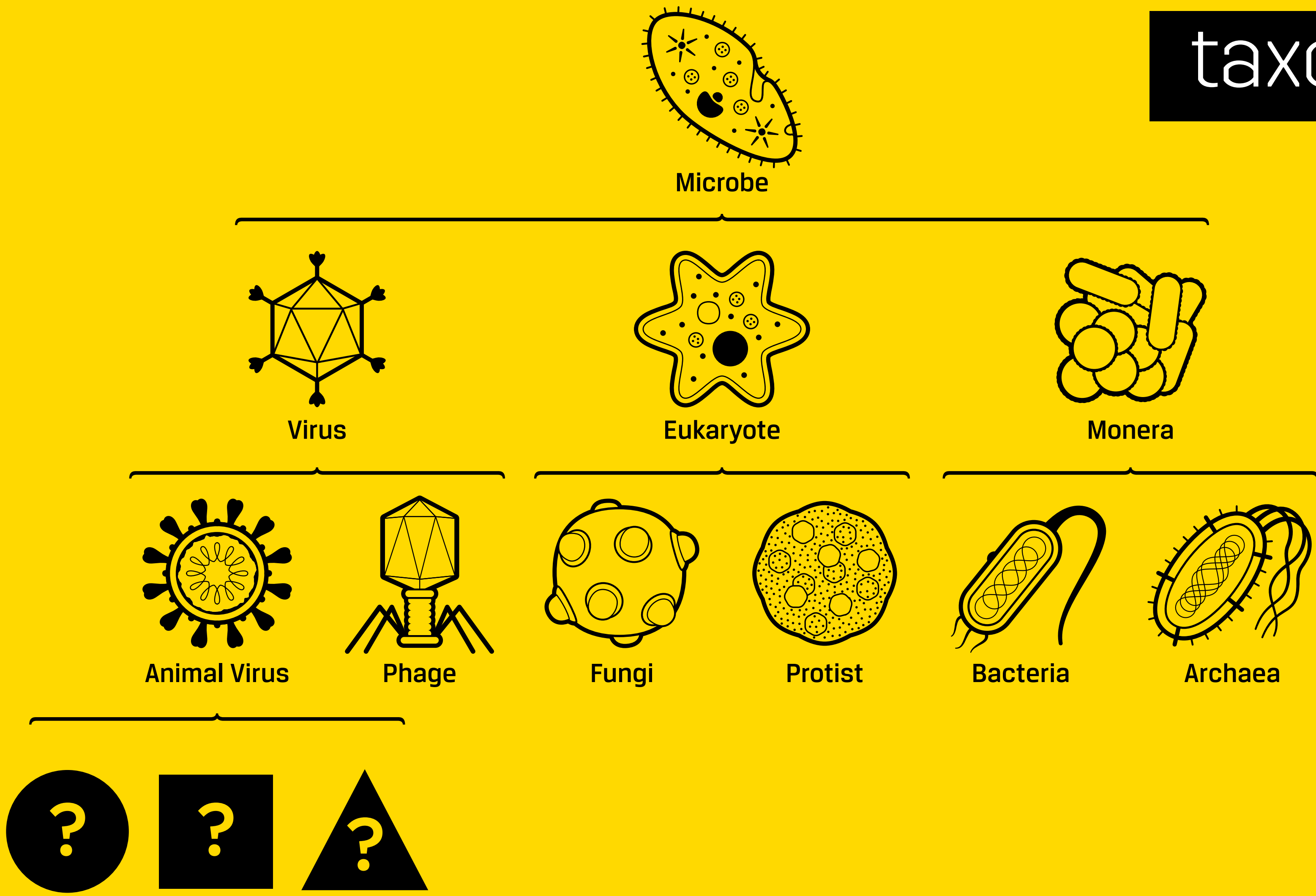
In the previous example, three systems were shown in an integrated map, with each system differentiated using color-coding. Other **graphical options** are available. Systems can be overlain atop one another, with some indicator of intersection, as is common for many other types of map (such as data maps). Alternatively, the systems can be presented side-by-side, with links of some kind showing the points of contact between systems. Or each system can be shown on a separate map, with points of contact indicated by cross-references ("jumps"), as is common in elaborate engineering drawings that take up multiple sheets of paper. In all cases, there is a value to making each type of system visually distinctive so that the layering is obvious. With an interactive display, it should be possible to add and subtract layers with a tap or click.

aggregation



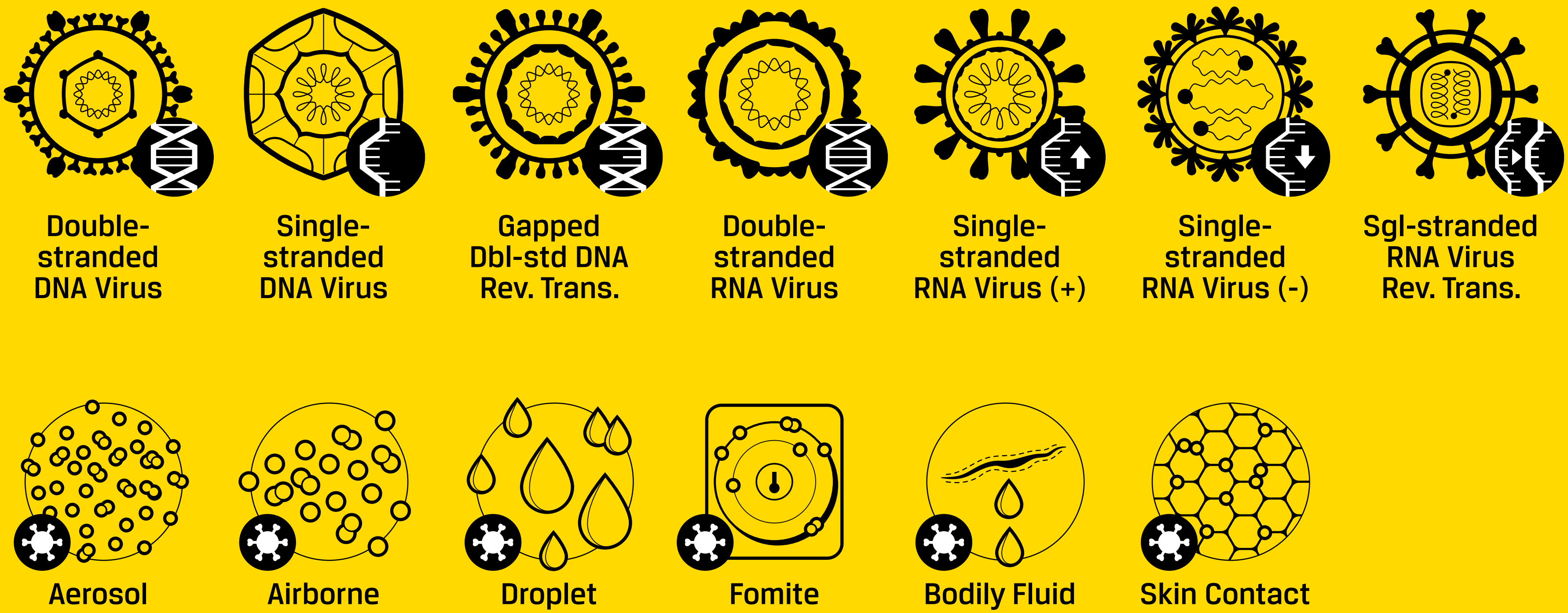
The second "A" in *Escalade* stands for **aggregation**. Aggregation is the extent to which like items are grouped together into kinds in order to summarize non-salient variation within the model. Highly aggregated maps are simple and tidy but *reductionistic* insofar as they conceal variation of potential interest to the viewer. One challenge is finding the level of aggregation most relevant to understanding the system or performing a task using the map. Disclosing that an item is an *aggregate* can be helpful, either to signal hidden variation or (medium permitting) allow the viewer to "drill down" through disaggregation. Another challenge is finding the right image to represent the group of underlying items, especially if that group is highly diverse. Not every aggregate comes with a handy cultural *archetype* nor will that archetype always be appropriate given the system under review.

taxonomy



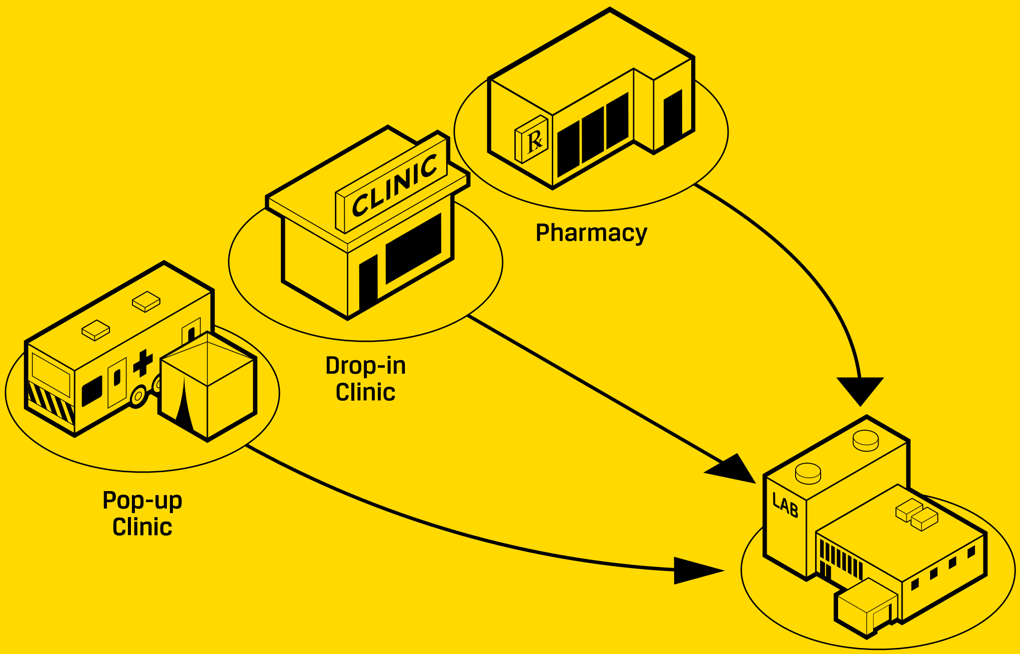
A third challenge is that the systems themselves often rely on classificatory schemes that formally group items. Or natural systems operate with *emergent kinds* that dictate how the system works. Scientists try to "carve nature at the joints," or create formal **taxonomies** that accurately reflect natural kinds. However, there inevitably is some culture influence to how items are grouped and labeled. For that reason, modellers should frequently revisit how systems create order by defining and enforcing the grouping of things into kinds, rather than just accept received categories at their face value. There is also more than one way to group items. Returning to viruses, a public health authority takes an interest in many different microbes that can potentially harm human health. But how does a testing system organize the category of thing called "animal virus?"

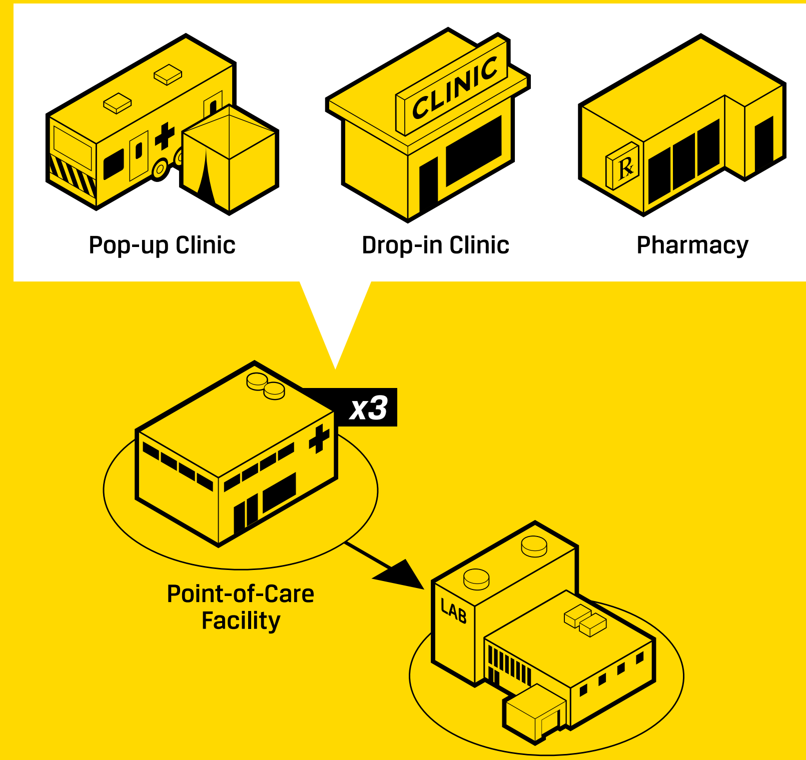
taxonomy



Different parts of the system tend to break down the category of "animal virus" down differently depending on their role. For example, makers of vaccines may be more interested in organizing viruses according to their genetic payload and receptor binding. Epidemiologists may be more interested in organizing viruses in terms of transmission types to find ways to limit spread. Infection testers may group viruses in terms of how they are detected. Thus, viruses are grouped according to whether they are discoverable by particular test methods. And pandemic managers may be more interested in how a virus of interest evolves into different sub-variants with different functional qualities. A system map should show the aggregates most relevant to particular processes within the system, instead of locking-in one classificatory scheme.

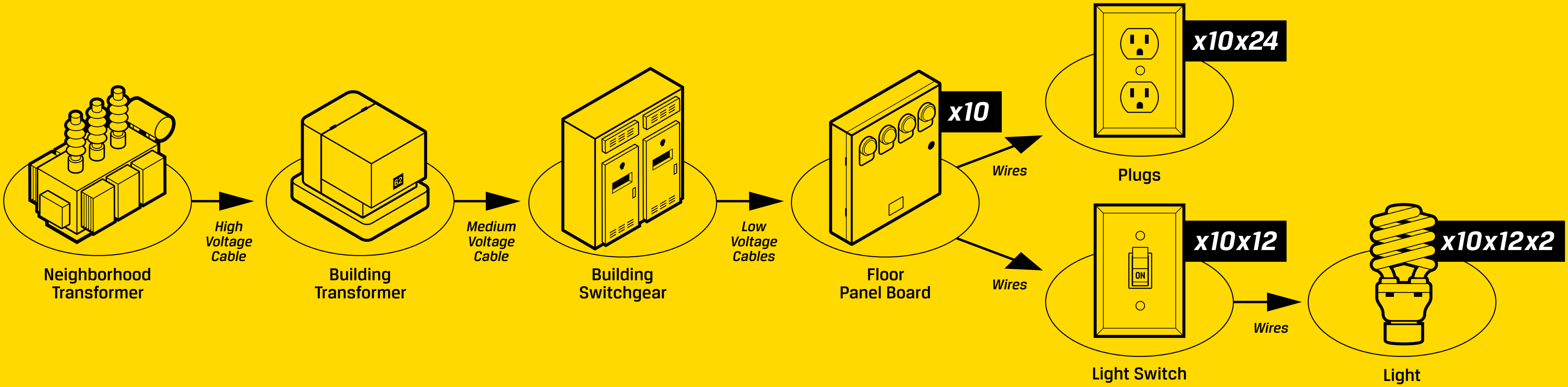
aggregate objects





Aggregation is also how maps are kept tidy. Within any system, there will be lots of redundancy and repetition. The point of the map is to summarize what is going on, not necessarily represent every instance of every object or process, especially if those instances are identical for all intents and purposes. For example, testing takes place at more than one kind of health facility. The specifics may not make a difference to the overall functioning of the system. Thus, multiple facilities can be collapsed into one with a summary label applied (possibly with some indicator of aggregation, such as number for underlying items). That can apply to multiple steps in the system too, providing there is a way to clue the viewer into the aggregation taking place. With interactive interfaces, selective revelation of disaggregation allows the viewer to "unpack" the aggregate to some degree.

aggregate strings

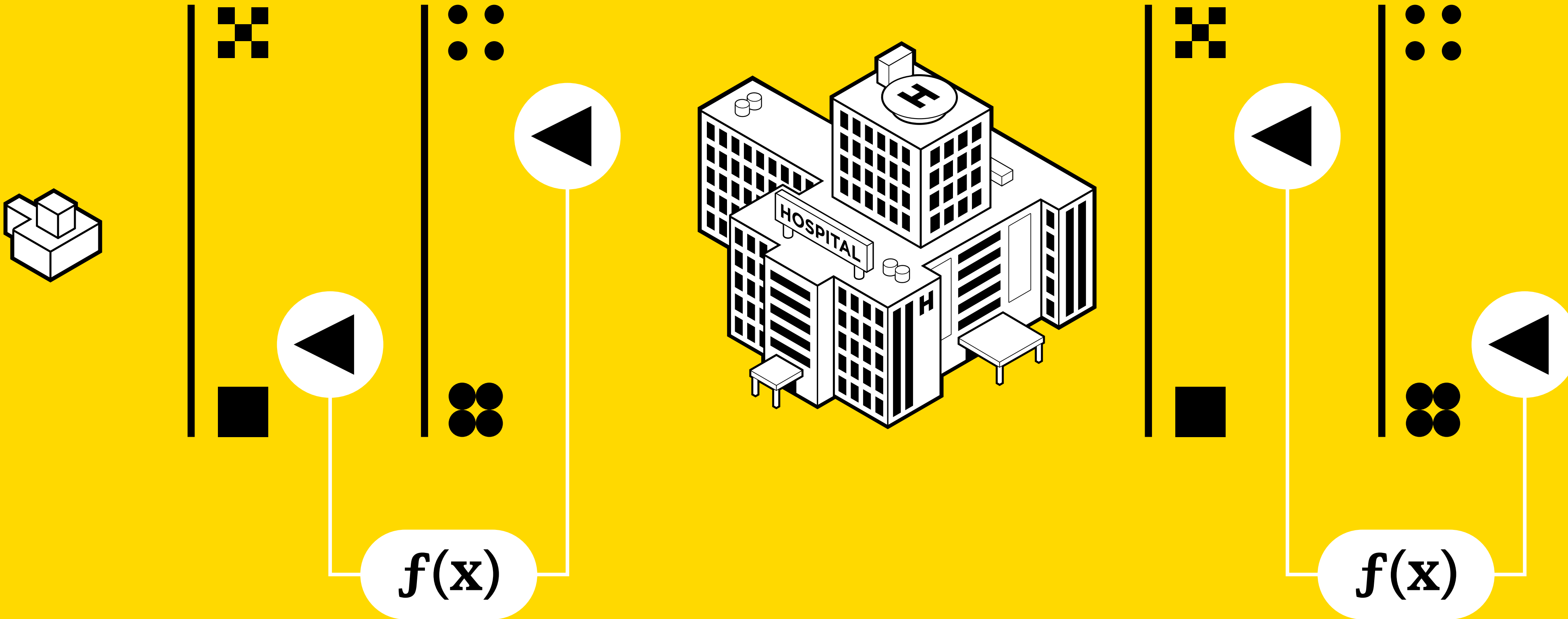


To drive home the point about aggregating system routines and sub-routines, consider the electrical system of a building. Drawing wires to every room would be redundant and make the map overly busy. Instead, multiple strings of links and nodes can be combined, with some indicator to alert the viewer to the aggregation taking place. For example, tags attached to objects or lines could indicate multiple levels of aggregation (x10, x10x24, x10x24x2). Or the aggregated segment could be bracketed and annotated, among other options. Note that most of the objects in the example are "black boxes," each one with inner workings that contain subsystems (such as electrical boards and wiring). In turn, those systems will contain objects with intricate inner subsystems (such as computer chips). These operate at different levels of scale and can be mapped accordingly.

detail

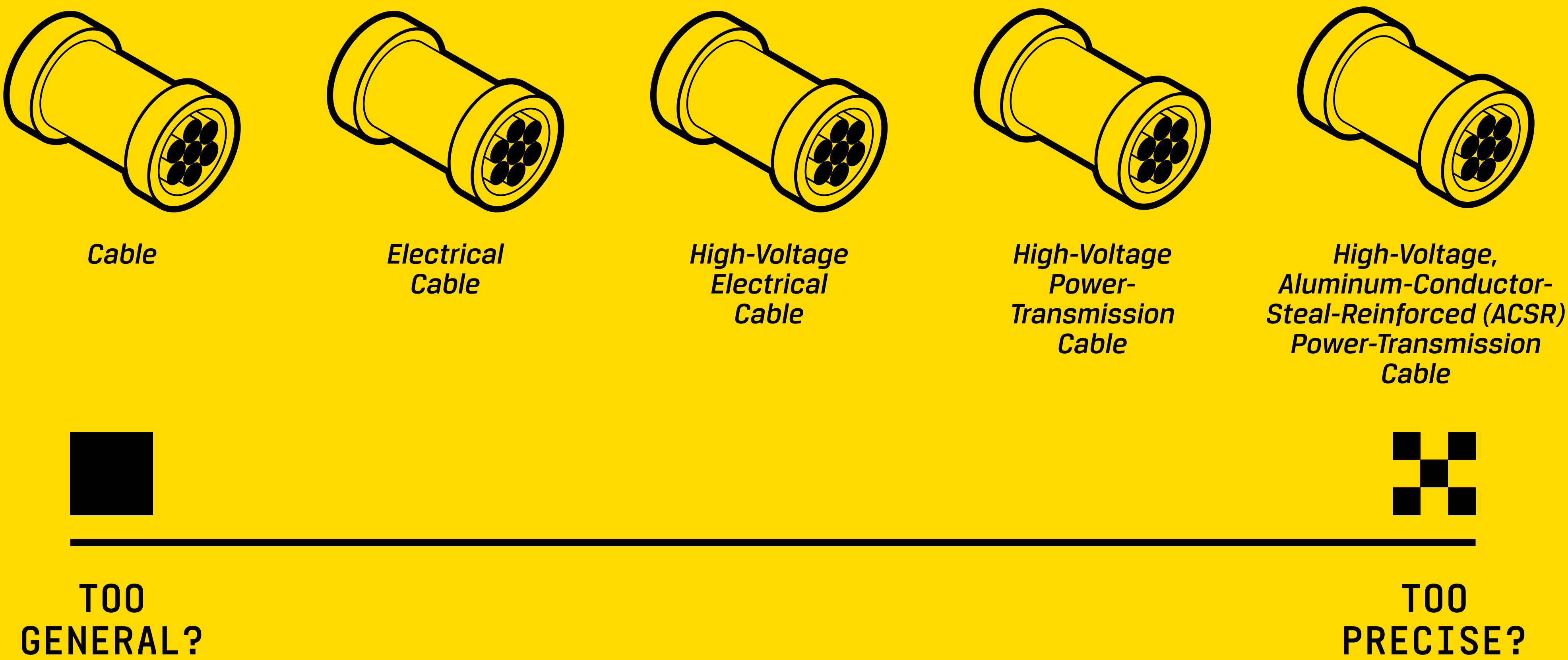
Next comes "D" for detail. **Detail** refers to the amount and intricacy of visual distinctions used to depict objects. High levels of detail improve realism at the risk of distracting features that add unnecessary busyness (*visual noise*). On the other end of the spectrum, too little detail results in amorphous objects that may be too nondescript. Also think about the map as a whole: too little detail makes it difficult to identify points of interest; too much detail can create cognitive overload, making the map hard to parse. Is there a sweet spot given the tensions? Is there just enough detail to make objects instantly recognizable? If only one image has to suffice, then an optimal amount of detail can be derived for most subjects and scenarios. Even so, we can do better by looking at how interactive interfaces handle detail dynamically as viewers zoom into and out of the map.

functions



Whether a particular level of detail is the "sweet spot" is an open question because that design decision often depends on other aspects of the map. For example, the appropriate level of detail depends on the treatment of size. Thus, in a sense, the *expanse* and *detail* dimensions are a **function** of one another. As objects grow in size, more detail may enhance the viewing experience, or at least not detract from it. It is common for automatically adjusting interfaces to rely on such functions, with zoomed in objects also revealing more detail. Other aspects of the *Escalade* model are tied together like this. For example, *scale* and *aggregation* often vary together. It may be easier to understand a system at high levels of scale if there is more summary grouping. That depends on the analytical goals of the viewer. Likewise with *aggregation* and the *abstraction* of images used.

textual detail

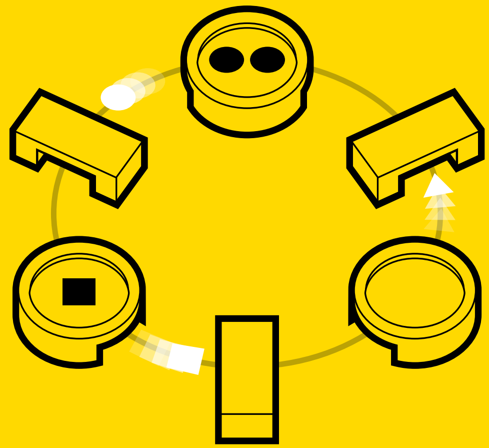


Detail also refers to text. How elaborate does a descriptive label have to be? For example, will "Cable" suffice to designate an electrical conduit? Or "High-Voltage Electrical Cable?" Or an elaborate technical designation?" Similarly, an annotation may be a brief summary or it may include many fine distinctions. How much **textual detail** is appropriate? Much depends on the space constraints, intended audience, and usage scenario of the map.

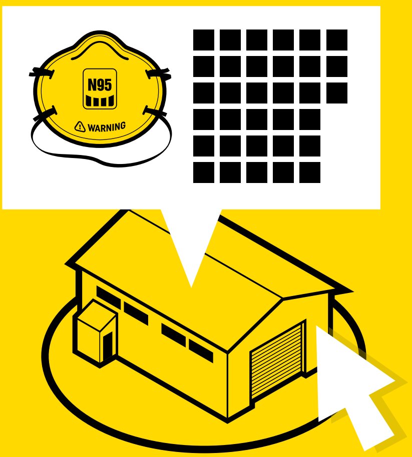
If a technician needs to know details to perform maintenance tasks, precise distinctions may have to be included (what make and model of cable?). If the map is for a general audience, technical details can be left out but orienting information may have to be added (what sort of cable?). Ideally, elaborations are hidden (or de-emphasized) for parsing the map but are made accessible (perhaps with a click) for other viewer tasks.

enlivenment

SOUND



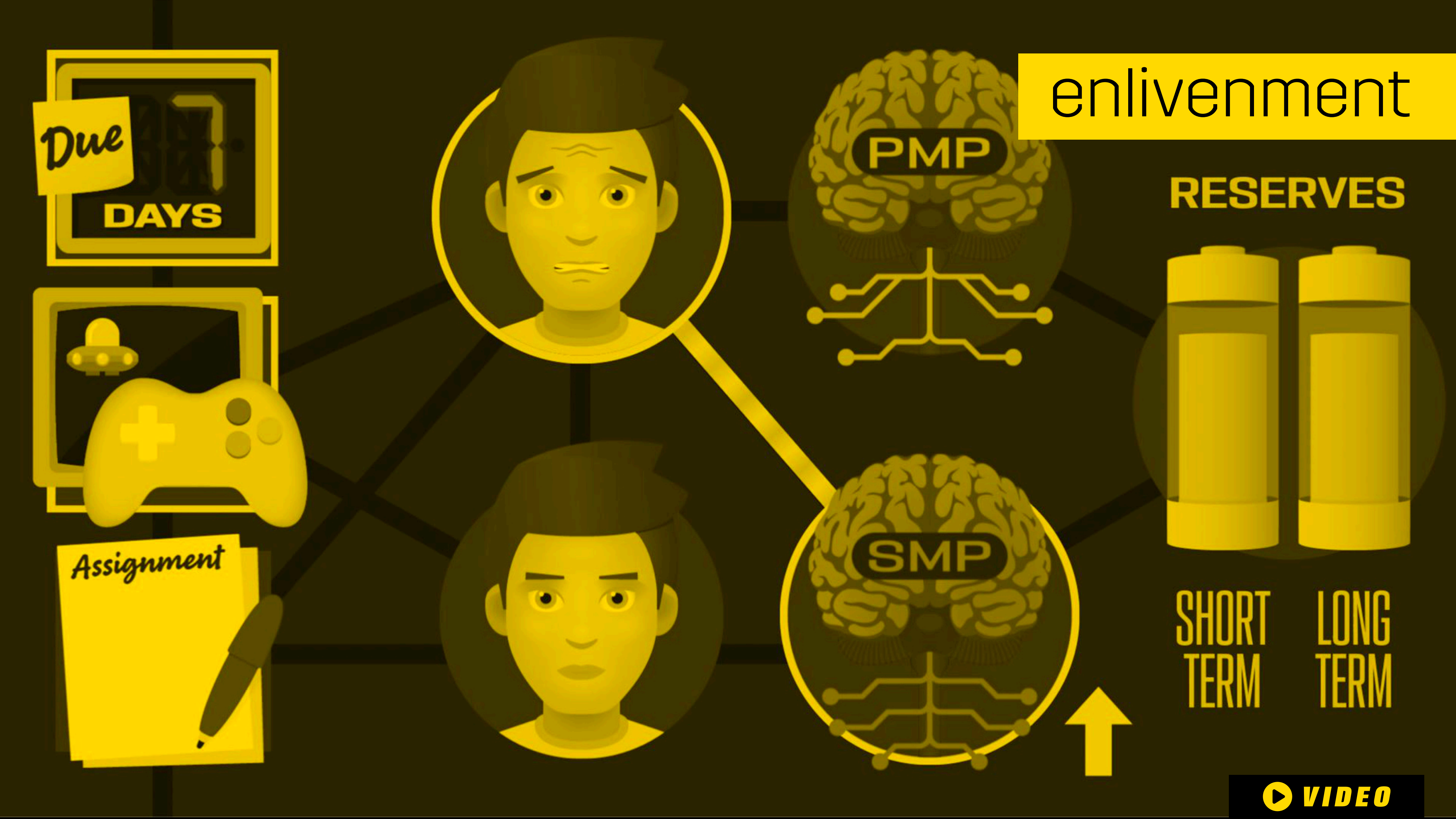
MOTION



INTERACTION

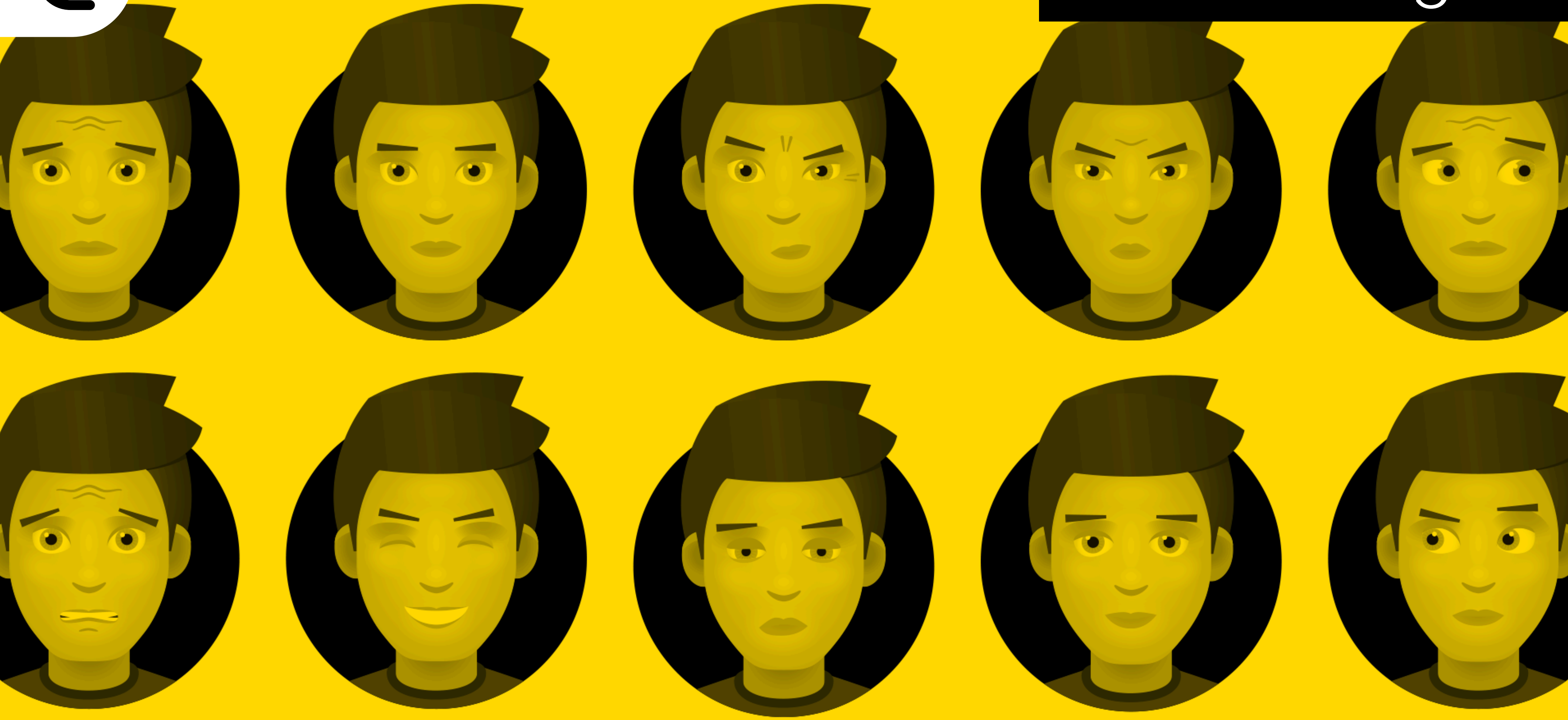
VIDEO

The last "E" in *Escalade* refers to **enlivenment**, or the introduction of sound, motion, and interaction into the map. High-resolution screens have become ubiquitous, whether on portable computing devices or large surfaces (tables and walls). System maps can become physical installations, such as those found in museums or system control-rooms. Virtual- and augmented reality technologies offer new opportunities to immerse ourselves in our models. All of these media benefit from the liveliness of graphical content, which brings other design challenges. Instead of adopting the usual lowest-common-denominator approach, the hope is that maps "degrade elegantly" to remain effective in static documents. That is a "liveliness first" method, akin to the "Web-first" and "MobileWeb-first" design movements that brought rich content to the Internet.



Here is a mental-health example from my own practice. I was asked by the Hart Lab at the University of Toronto to visualize the *Uneasy Modulation Model*.* The model shows the stress caused by the interaction between situational factors, avoidance behaviors, the nervous system, and energy reserves. The video version is a motion graphic that teaches viewers the parts of the model before presenting the whole thing on one screen. Three scenarios are presented. An overdubbed narration tells the big-picture story while furnishing important details that are not illustrated. Moreover, a black-and-white static version was made for the scientific paper.

* Josoph Arpaia and Judith P. Andersen, "The Unease Modulation Model: An Experiential Model of Stress with Implications for Health, Stress Management, and Public Policy," *Frontiers in Psychiatry*, vol 10, art. 379 (2019). The article and full video are available by clicking here.

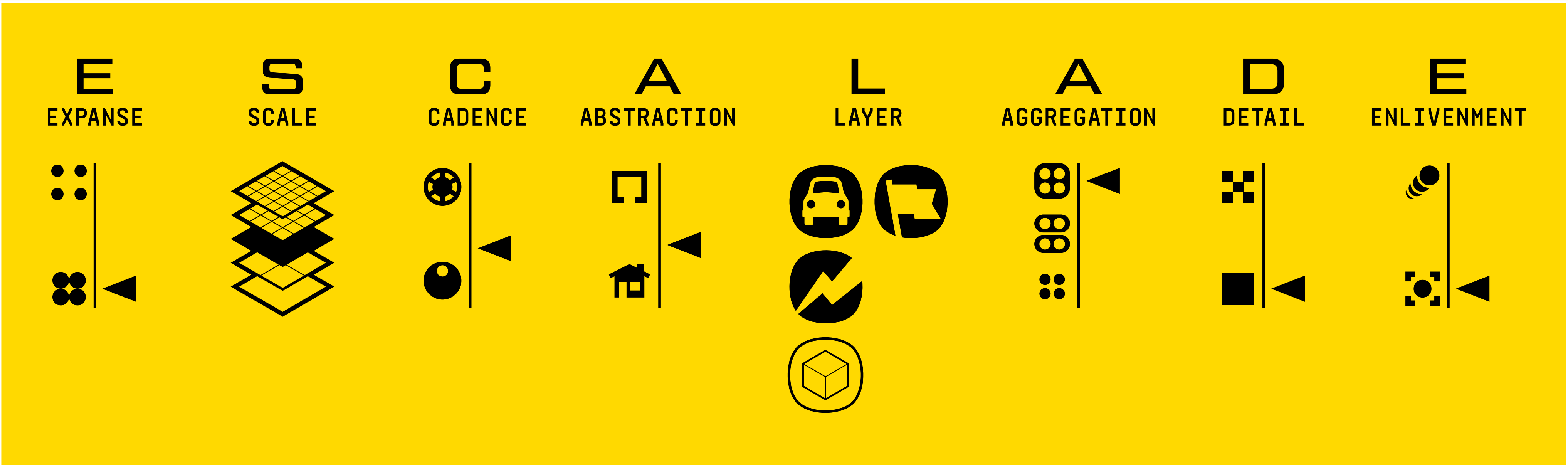


state changes

What benefits does the enlivened version offer over the static one? The sequencing of complex interactions can play out, making it easier to understand system dynamics. That is done by showing animated links that imply the effect of one node on another. Moreover, nodes can change in response to that effect, signalling a state change. For example, in the graphic, changes in facial expression show changes in a person's emotional state, either in terms of amount of anxiety or task difficulty. Facial expressions are an indirect indicator but are good at showing emotional states given that humans are adept at inferring emotions from faces (albeit in a culture-specific way, in part). In this case, the contextual variables (left column, previous slide) can also be swapped out to explain another scenario and show how situational variables can change system dynamics.

framework

DASHBOARD

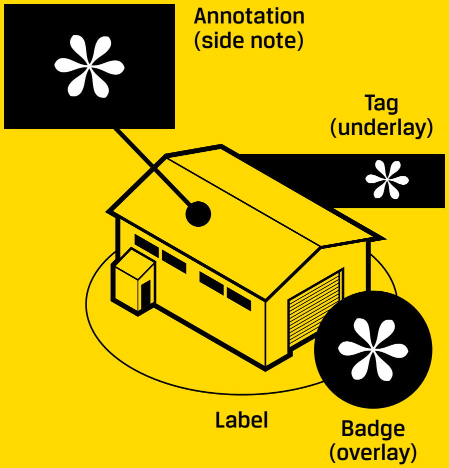


That completes the review of the various dimensions of the *Escalade* framework. Admittedly, there is a lot going on with each one; that is, each dimension has several underlying considerations. Nevertheless, the framework serves as a handy guide for any system mapping activity involving notations. Existing notations fall short on several counts. Most are highly abstract and spatially aloof. They run rough-shot across scales, types of system, and levels of aggregation. Insofar as pictures are used, they tend to be generic icons and clip-art that are a poor fit. There is a lot of low-hanging fruit for notation improvement. As the foregoing discussion of graphical-device options suggests, however, there is also an opportunity to reimagine what system notations can be at a more fundamental level.

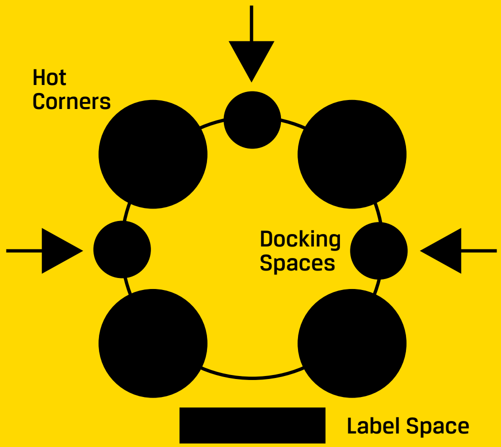
reductionism

Every manipulation using the *Escalade* framework entails some loss of information. That is a feature not a bug, otherwise the map would equal the territory. That said, crucial information should not escape our attention or otherwise fall off the map, like liquid slipping through our cupped hands despite best efforts. Clumsy **reductionism** is what can potentially mislead those using system maps to make important decisions. The *Escalade* framework can be thought of as a way of keeping track of what information gets included or excluded. That is a crucial function of an information-design framework. For example, manipulations associated with expanse will cause a loss of distance or size information. If that information is important for viewers' analysis of the system, then it has to be added back in somehow (or made easy to retrieve). How would that work?

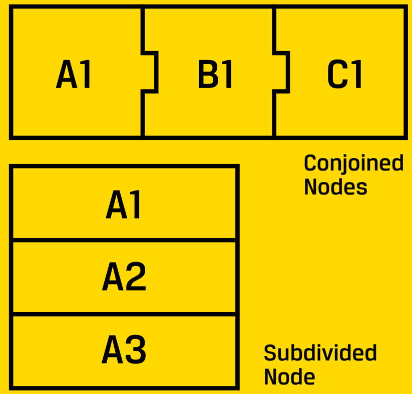
graphical devices



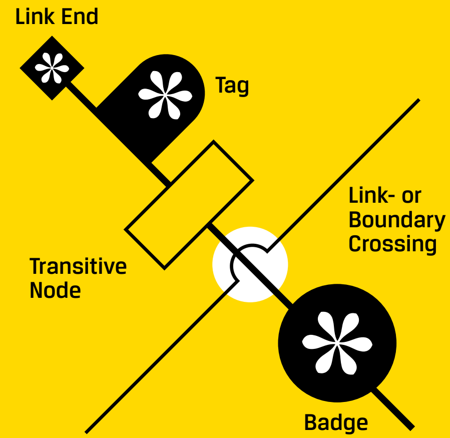
AUGMENTED NODES



NODE ZONES



NODE DIVISIONS



AUGMENTED LINKS

Escalade is a step towards a new notation by framing challenges and setting evaluation criteria. The second step is the *Anatomy of System Notations*, a comprehensive inventory of graphical devices used by existing notations.* All of the conventions and niche experiments are itemized. Several devices taken from that inventory show a range of options for adding information back into a map. For example, a node may be overlain with a *badge* on one of the four corners. It can show a pictogram, encoded symbol, or alpha-numeric string. The number of options for selective revelation of information multiplies with interactive interfaces. The point is to encode one type of information with each device and do so consistently.

* To be released as a presentation and poster at the Systemic Design Association's *RSD12 Symposium* at Georgetown University, Washington, DC, October 18-20, 2023.

project road map

To understand where the *Escalade* framework is going, consider the SystemViz project roadmap. A new notation is an important goal. That notation will be **"opinionated"** rather than try to be "all things to all people" in a ways that satisfy no one. In other words, any design project involves tensions and trade-offs. Inevitably, sacrifices will be made to strike a balance. With the forthcoming notation, those judgements come with defensible priorities grounded in information-design principles, a motivating task-loop, and realistic usage scenarios. For example, the current default is an "everything, everywhere, all-at-once" approach to mapping, one which is self-defeating: everything is included so that nothing is missing, only to have items get lost in the resulting mess. There are better ways. And new technological media add options for making intelligent trade-offs.

thank you

SYSTEMVIZ

www.systemviz.com

Further progress will be documented on the SystemViz web site. As with other tools offered there, the *Escalade* framework is offered in the spirit of open science, with an open-source license. Anyone can use and modify the framework with attribution. I encourage you to sign up for e-mail updates on the site. Thank you for your attention. And many thanks go to IIID as generous hosts. If you have any questions or comments, feel free to send those along. This is the first airing of the proposed framework, with the framework an early step towards creating a new notation. There is plenty of opportunity to influence the trajectory of this project and I encourage you to reach out.