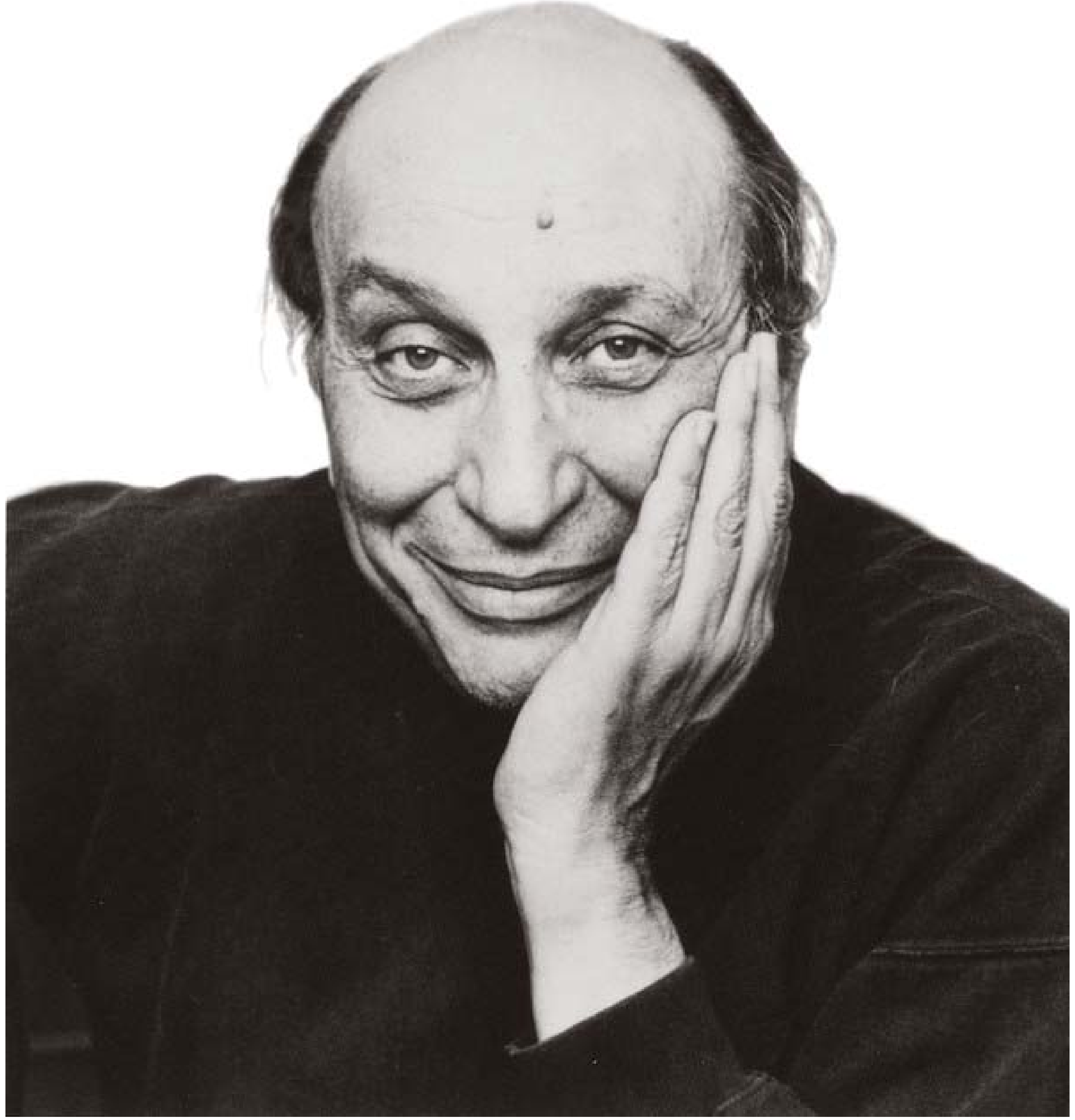
Accenture / Fjord   Dublin (via teleconference)   30 June 2020

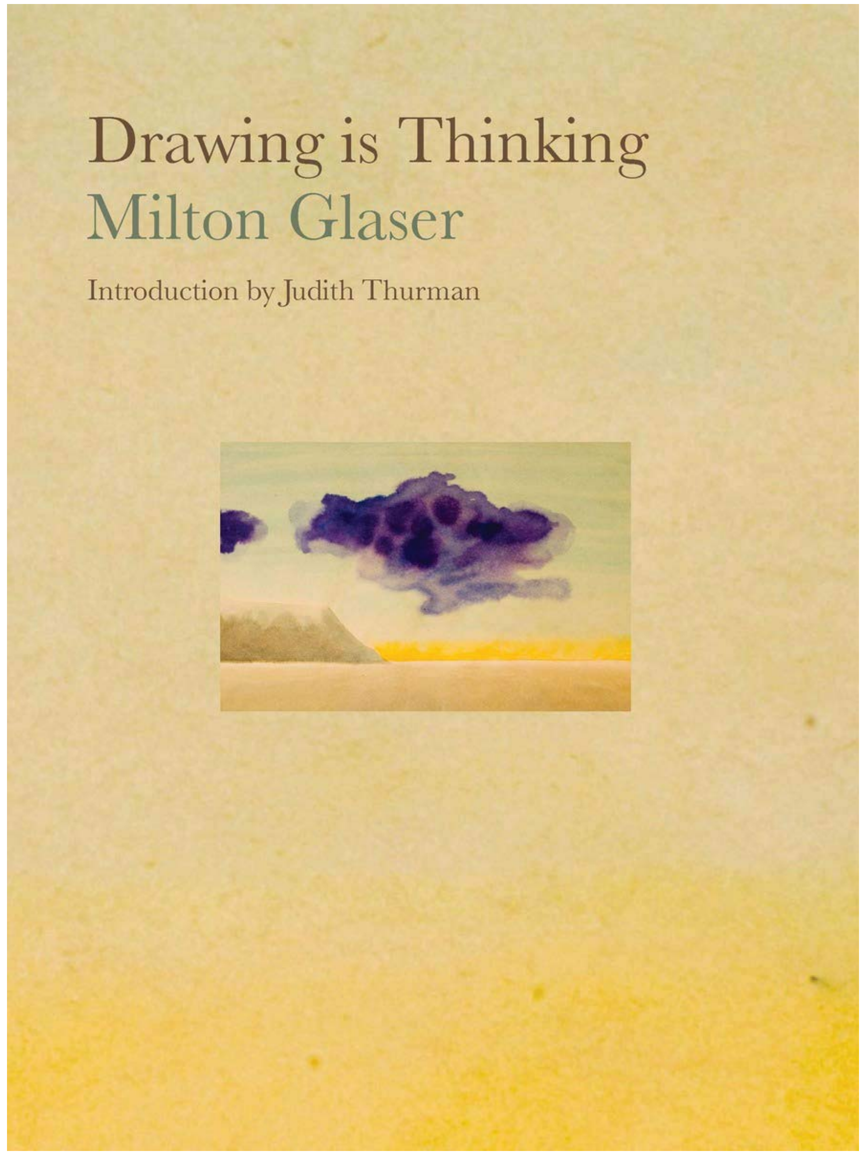
Systems Theory in Design

Models

Hugh Dubberly
Dubberly Design Office

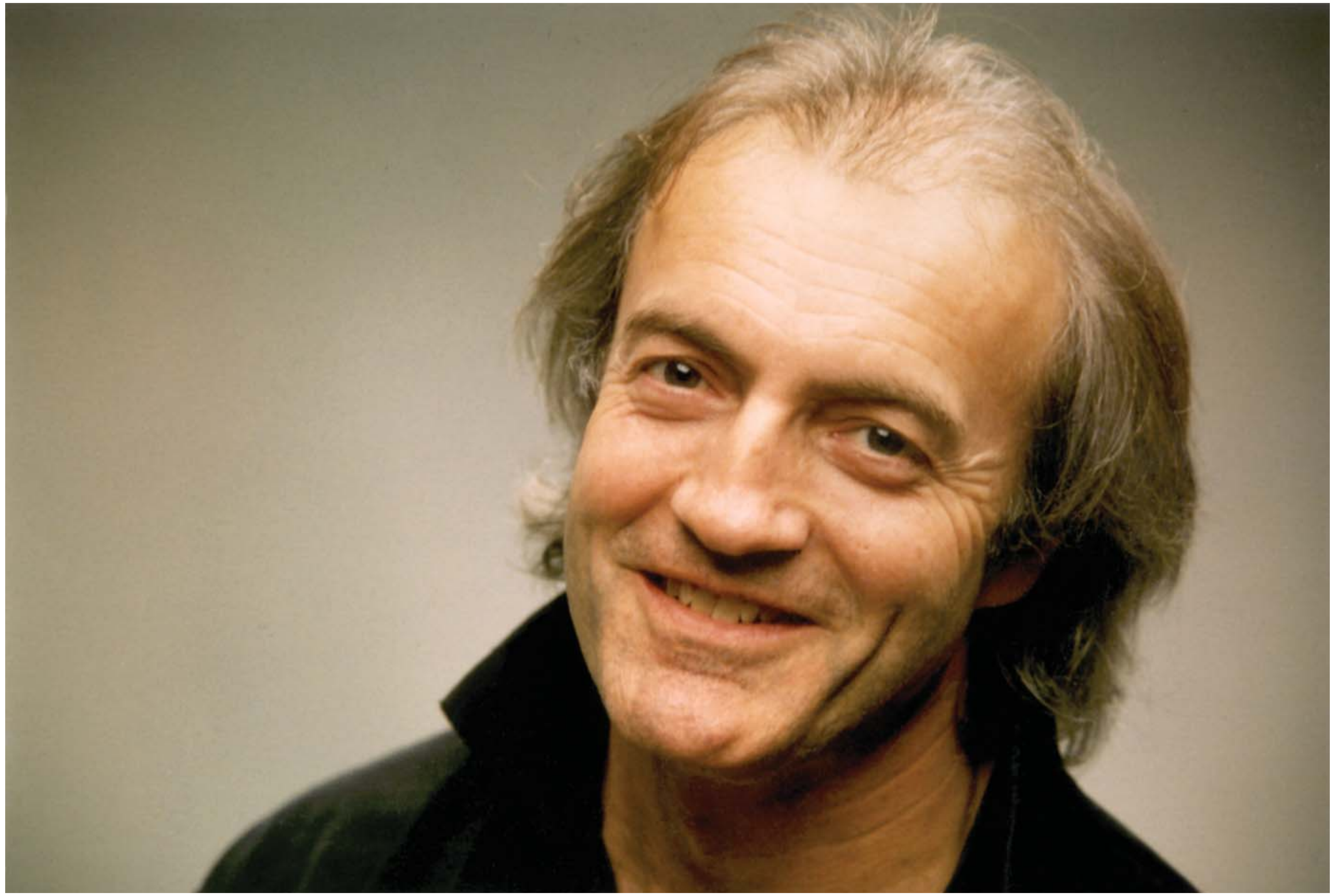
On Friday, Milton Glaser died — one of the last of a 20th century design heros. Glaser believed "drawing is thinking" and likewise that designing is drawing.





Drawing is Thinking
Milton Glaser
Introduction by Judith Thurman

Renowned computer scientist Bill Buxton believes something similar—that drawing is at the heart of designing.



"Bill Buxton brings design leadership and creativity to Microsoft. Through his thought-provoking personal examples he is inspiring others to better understand the role of design in their own companies."
Bill Gates—Chairman, Microsoft Corp.

Sketching User Experiences

getting the design right and the right design

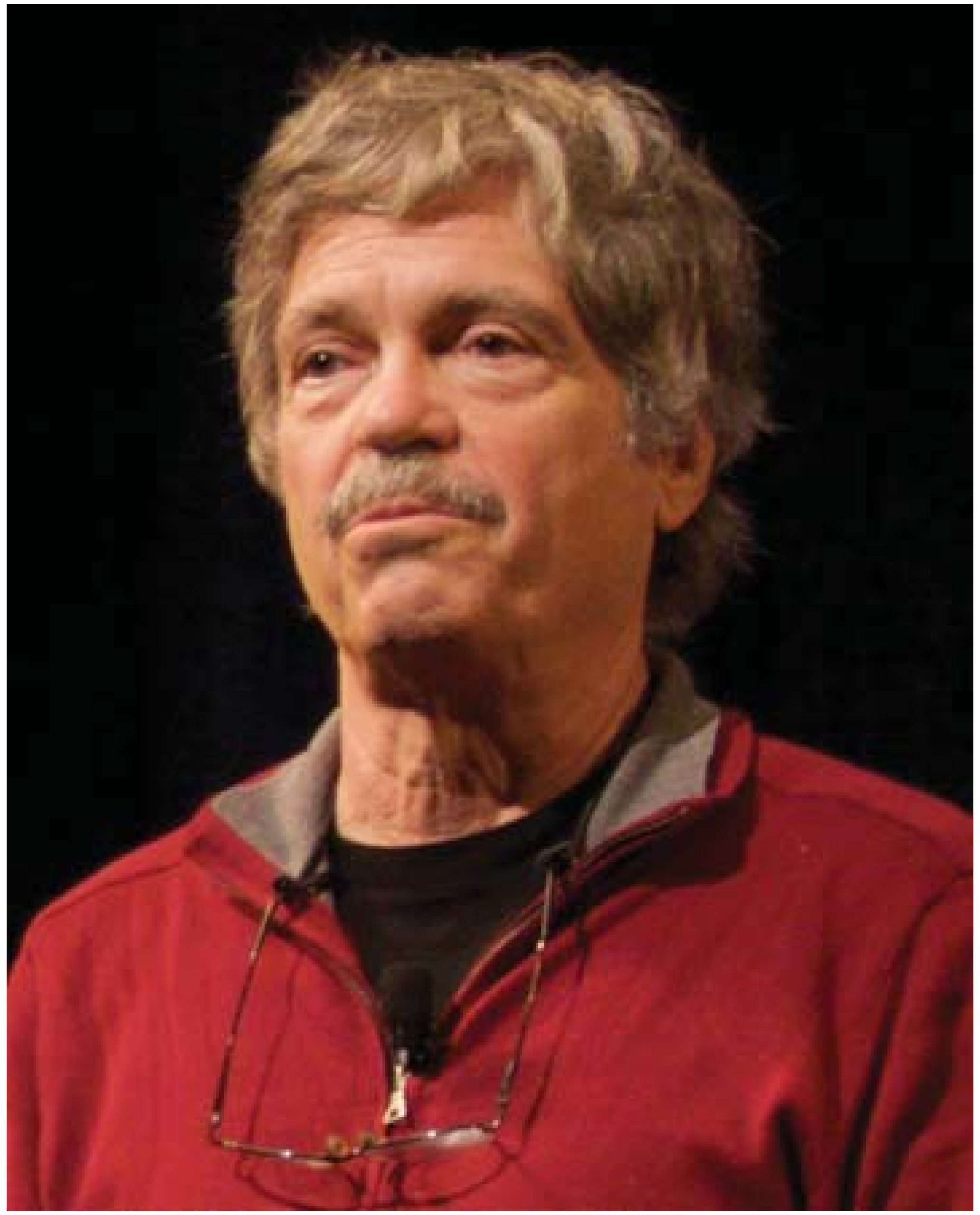
Bill Buxton

The role of drawing in designing is straight-forward for physical artifacts (products, communications, environments).
Drawing enables rapid cycles of iteration —
conversation with the situation and with stakeholders.

But systems are often intangible, even invisible.
They unfold over time and space.
Rarely do we have a vantage point from which to see their entirety.

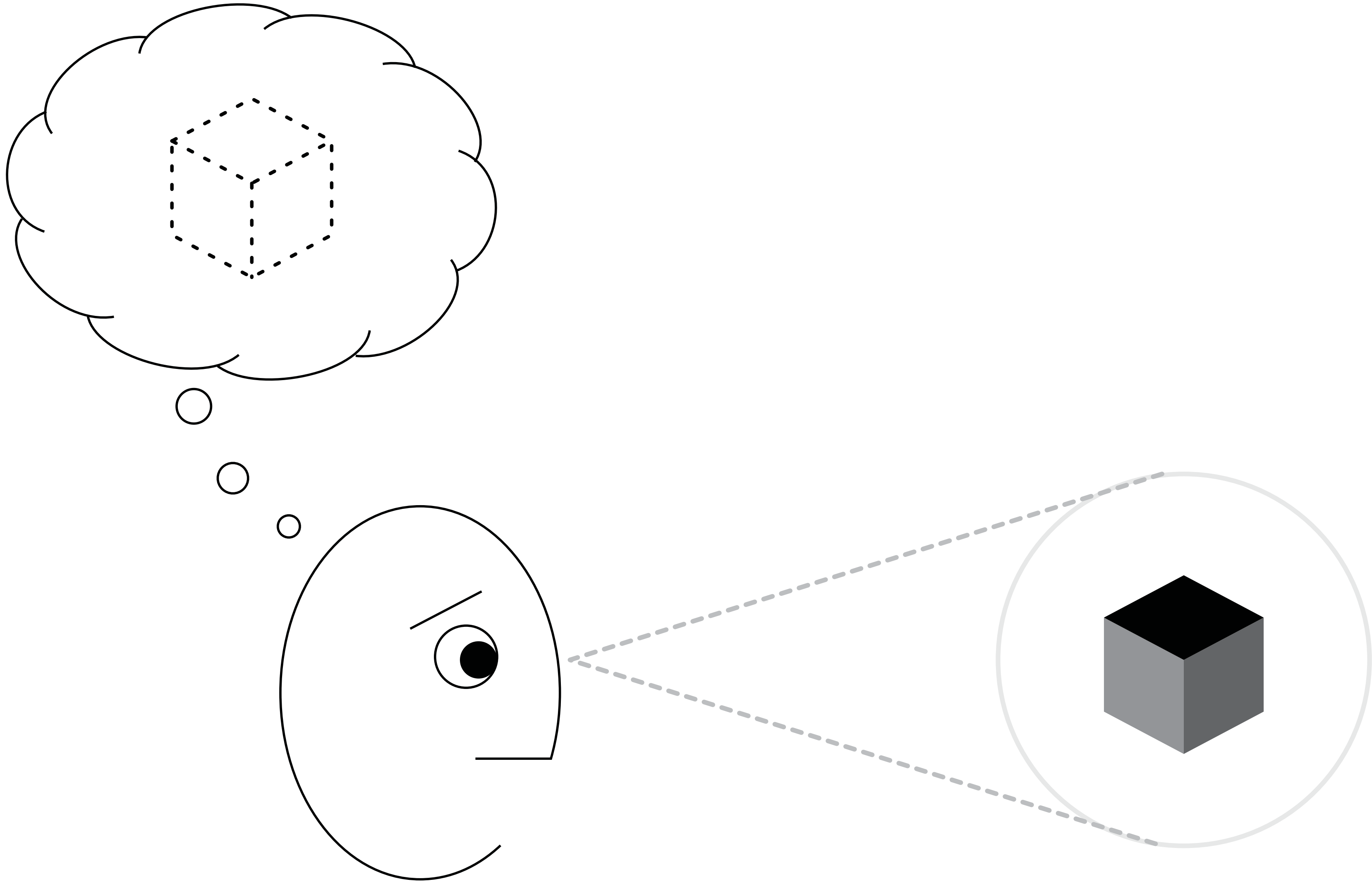
For systems, we require a new kind of "drawing" —
a way of representing systems
so that we can analyze them, propose changes, and discuss them.
So: We turn to "modeling".

“We do most of out thinking with models... And these models are our voodoo dolls.” –Alan Kay



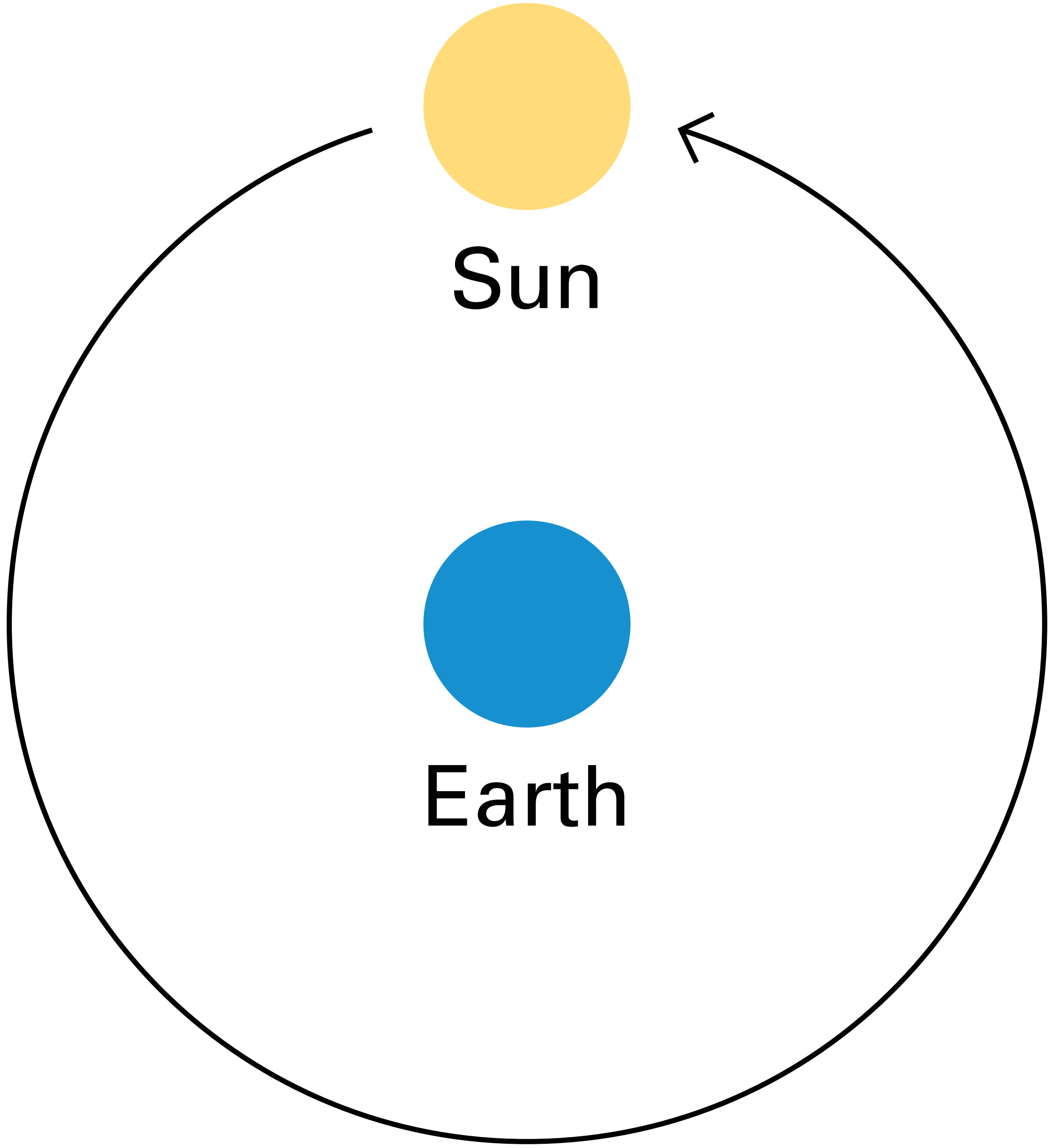
A definition + examples

A model is an idea about how part of the world works.



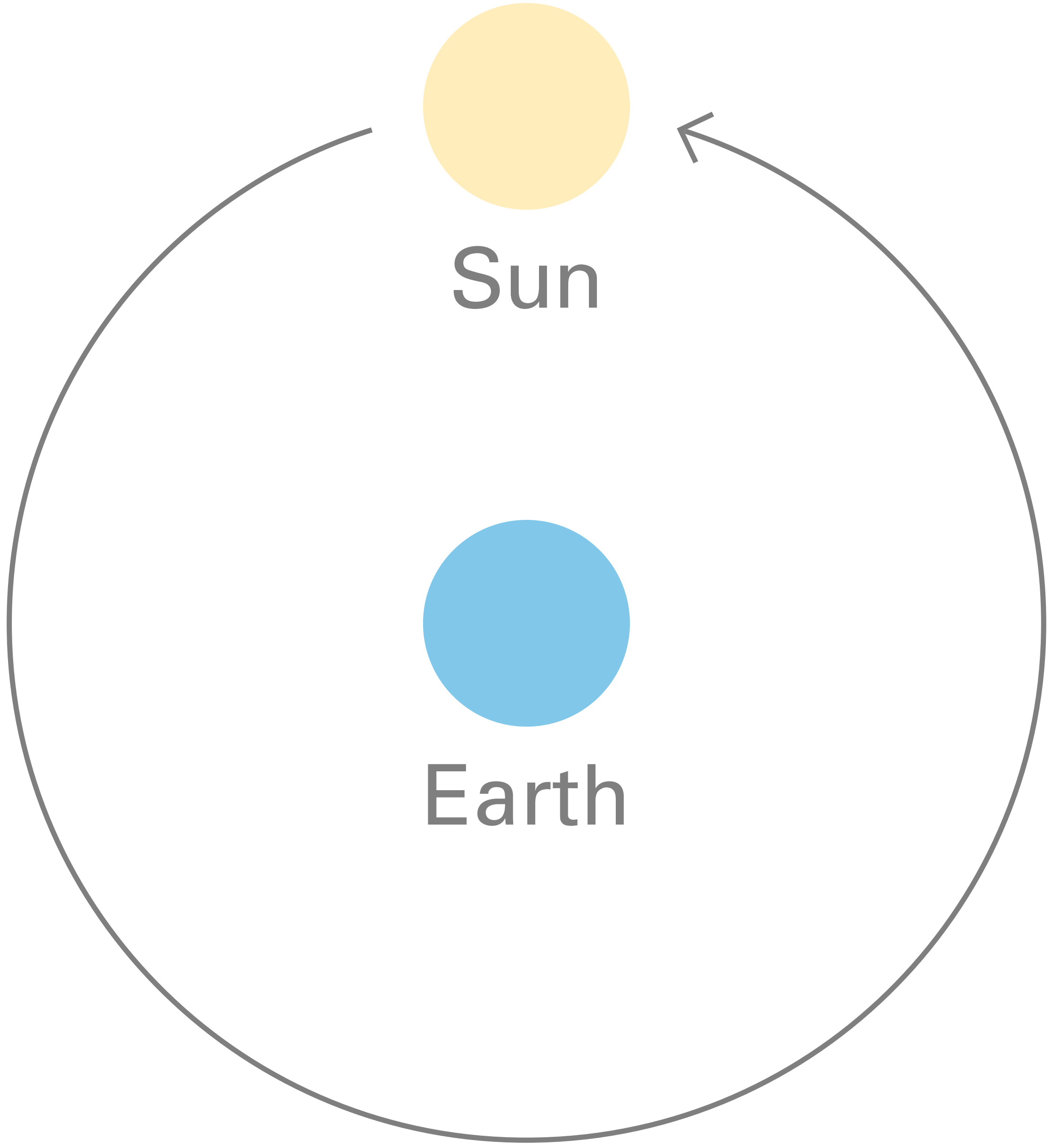
For example, we see the sun rise in the east and set in the west.

The apparent motion of the sun suggests this model.

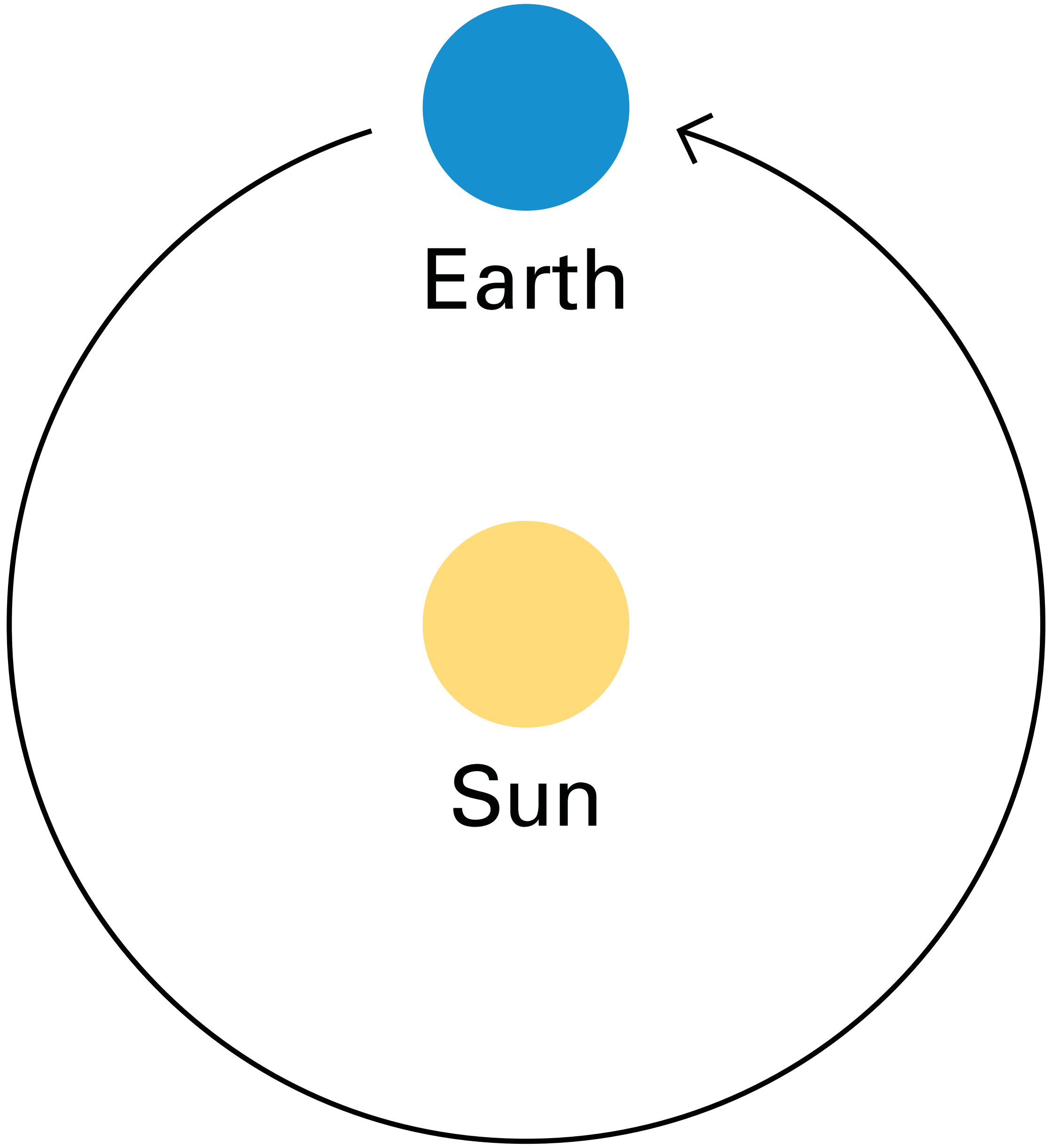


—Ptolemy

Despite what we see everyday, we think of the earth as revolving around the sun. Why? What observations support this model?



—Ptolemy



—Copernicus

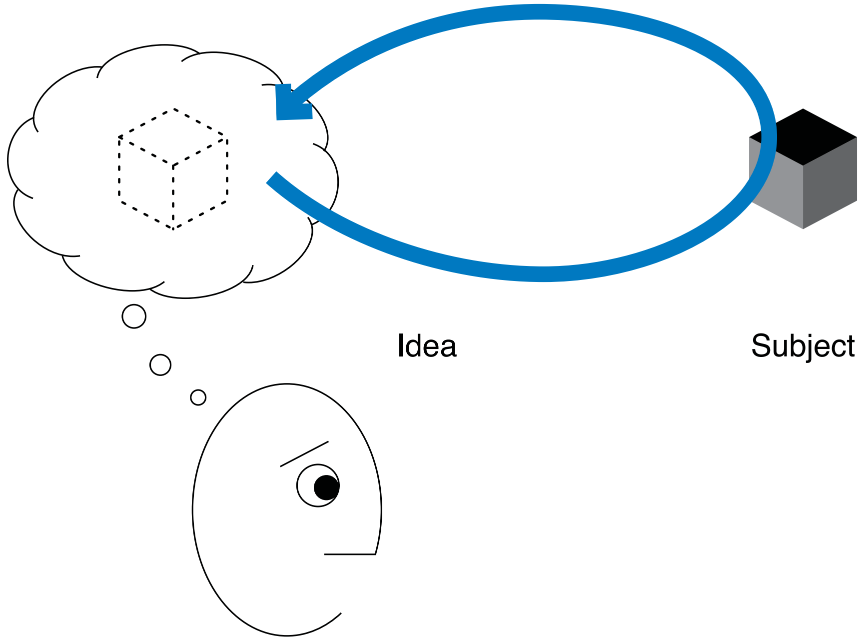
Mars sometimes appears to travel backwards; both Ptolemy and Copernicus explain Mars' retrograde motion, but the Copernican model is much simpler.

Another example, this x-ray photo taken in 1952 by Rosalind Franklin...

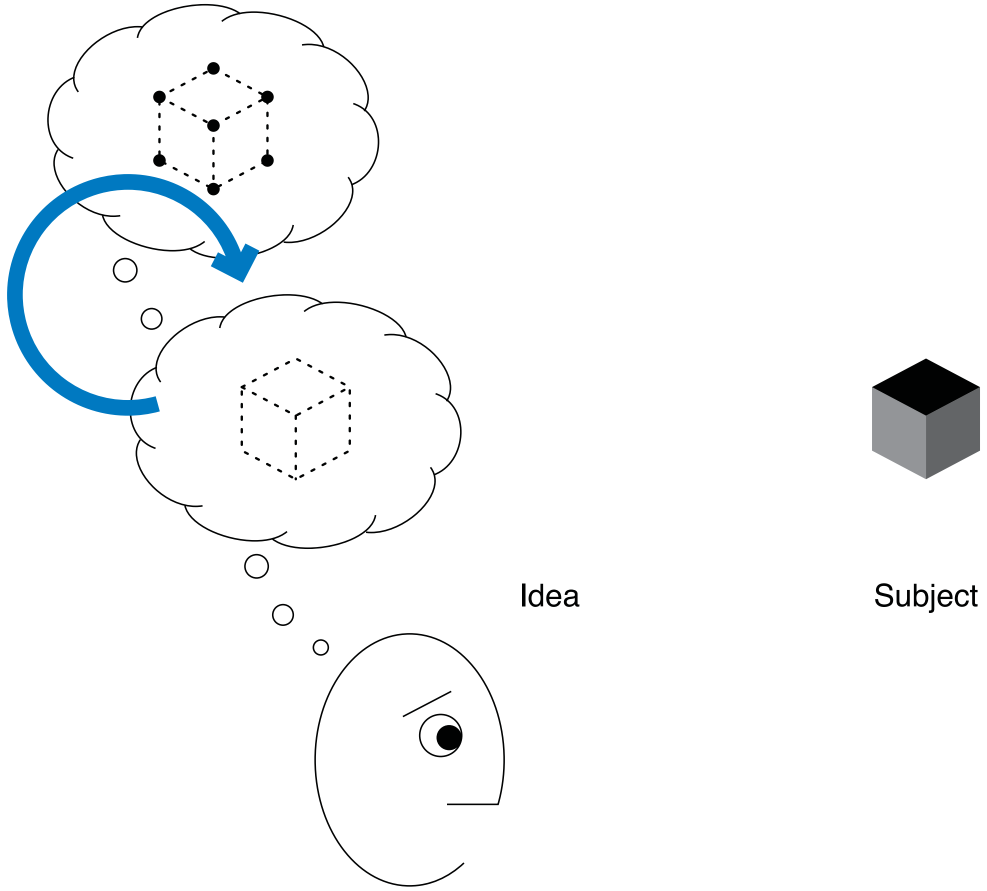
Another example, this x-ray photo taken in 1952 by Rosalind Franklin... **aided the development of our model of DNA.**

Models of models

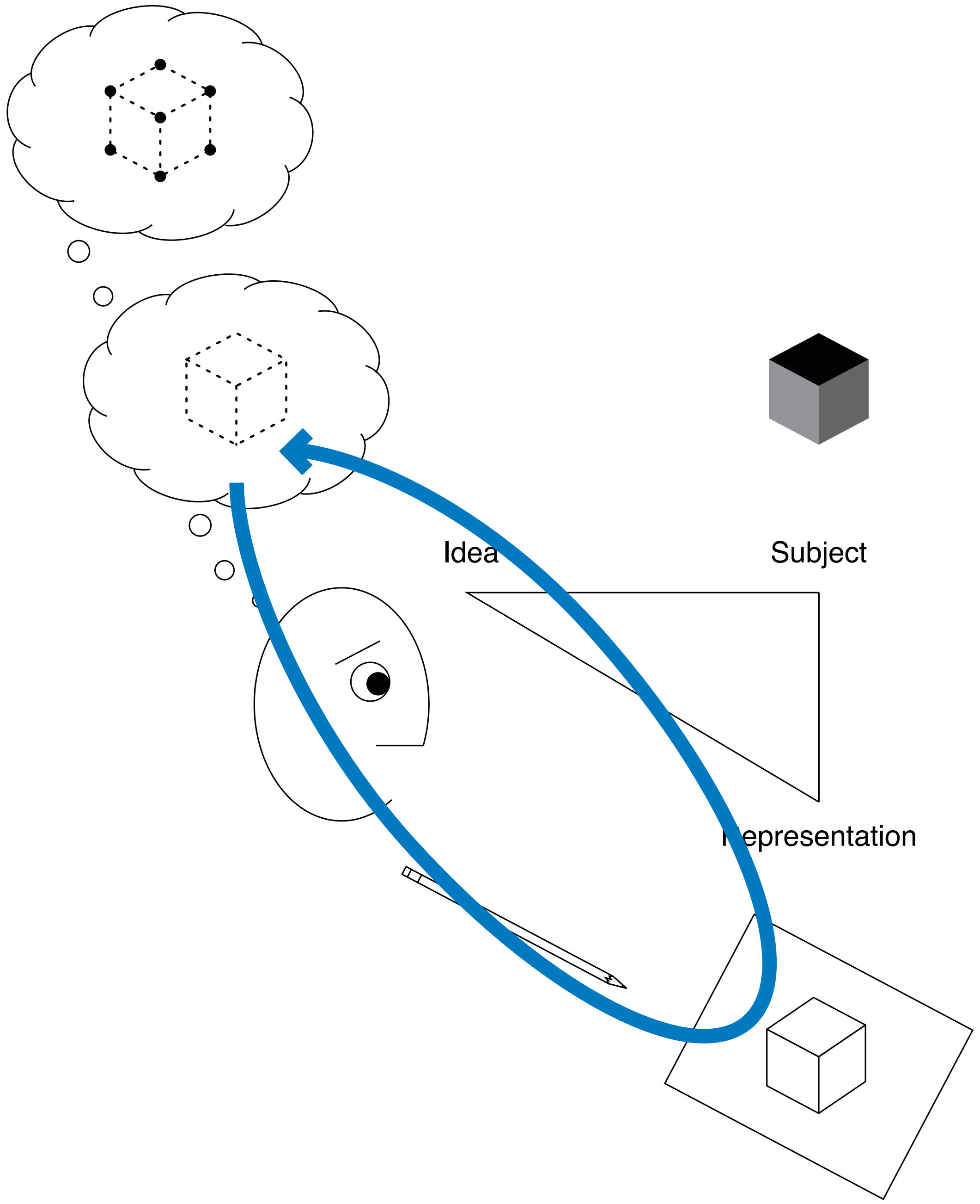
Models begin with observation and thinking.



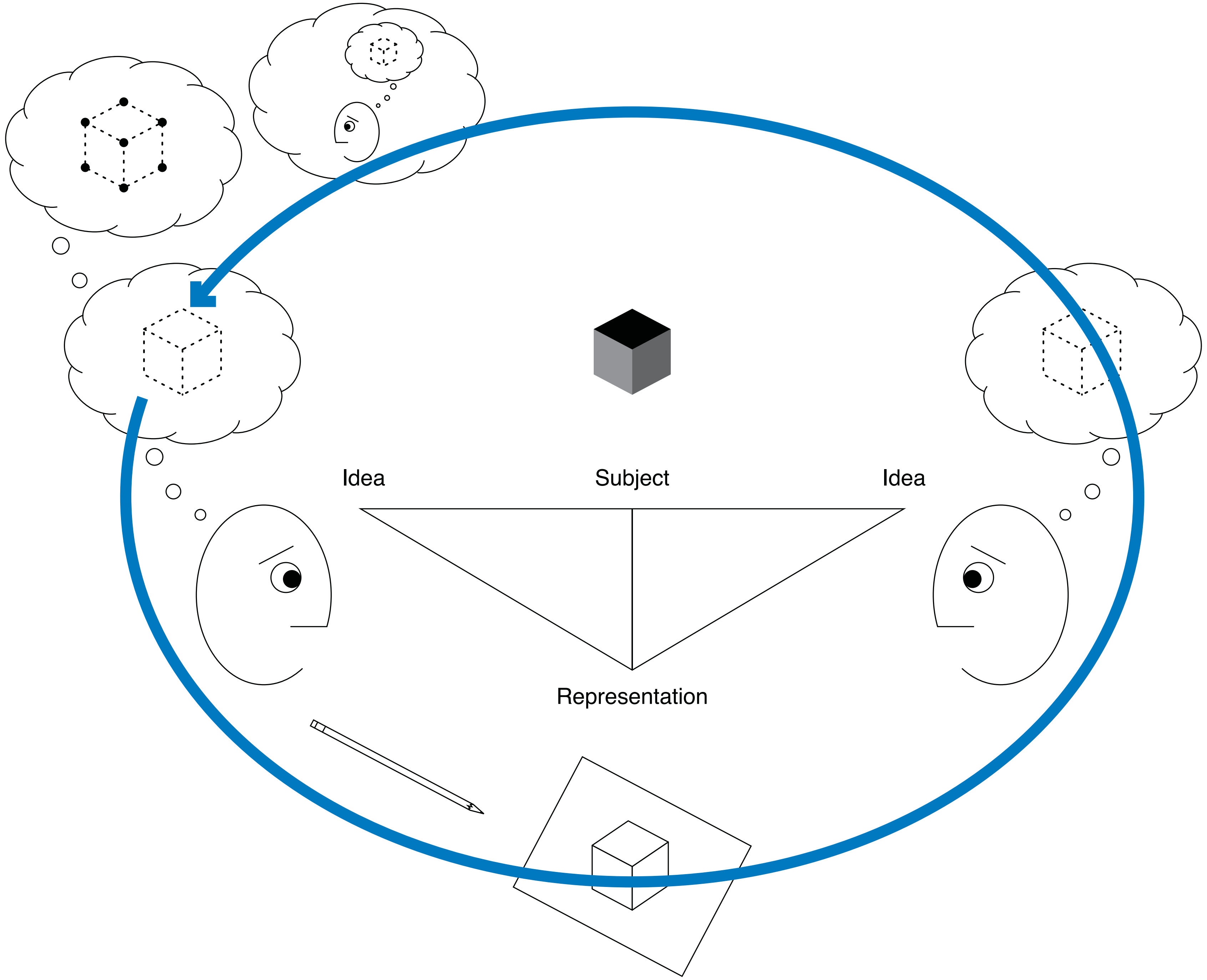
Thinking about how to explain our observations may lead us to think of alternatives—or related ideas.



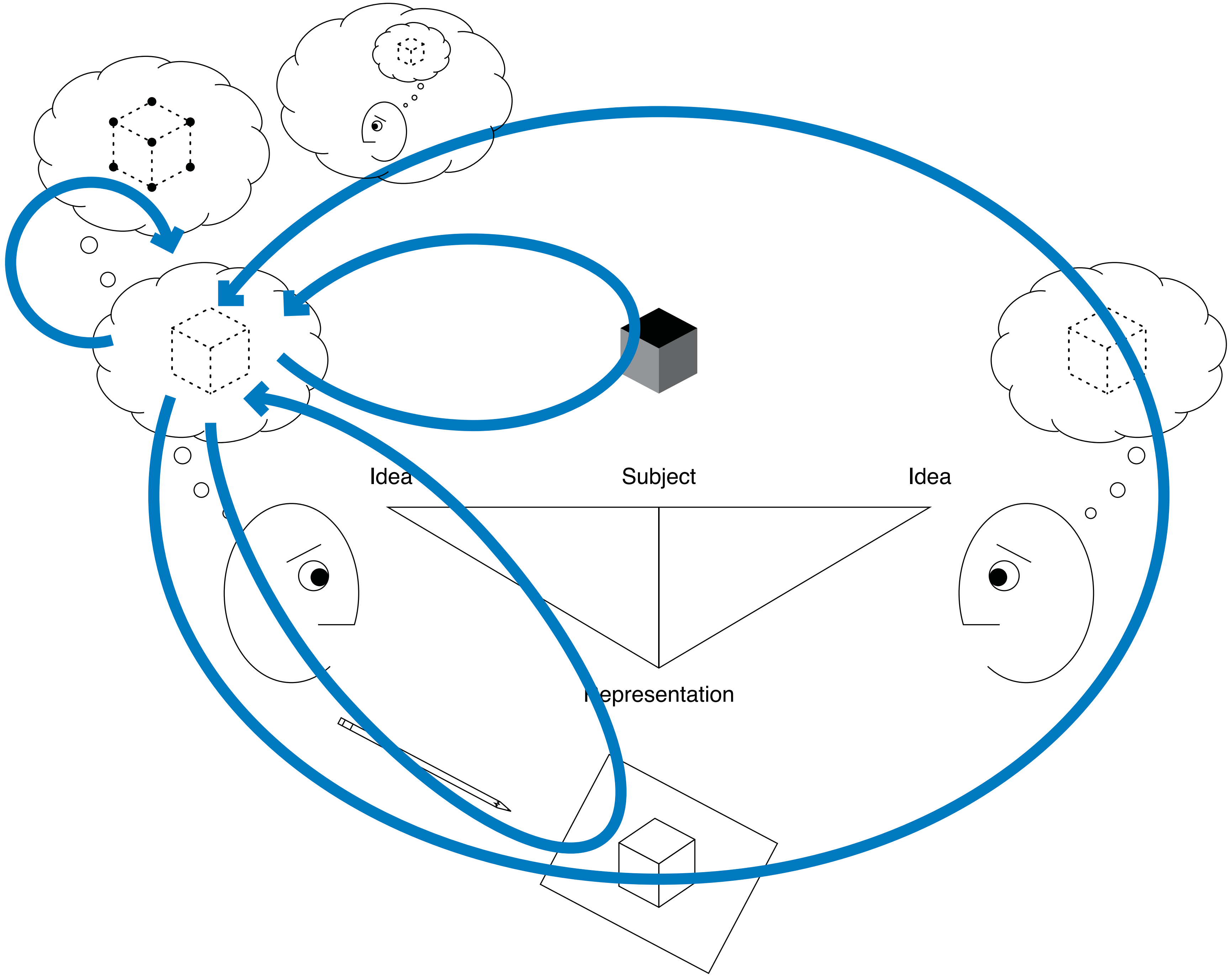
The process of representing an idea may change the idea itself.



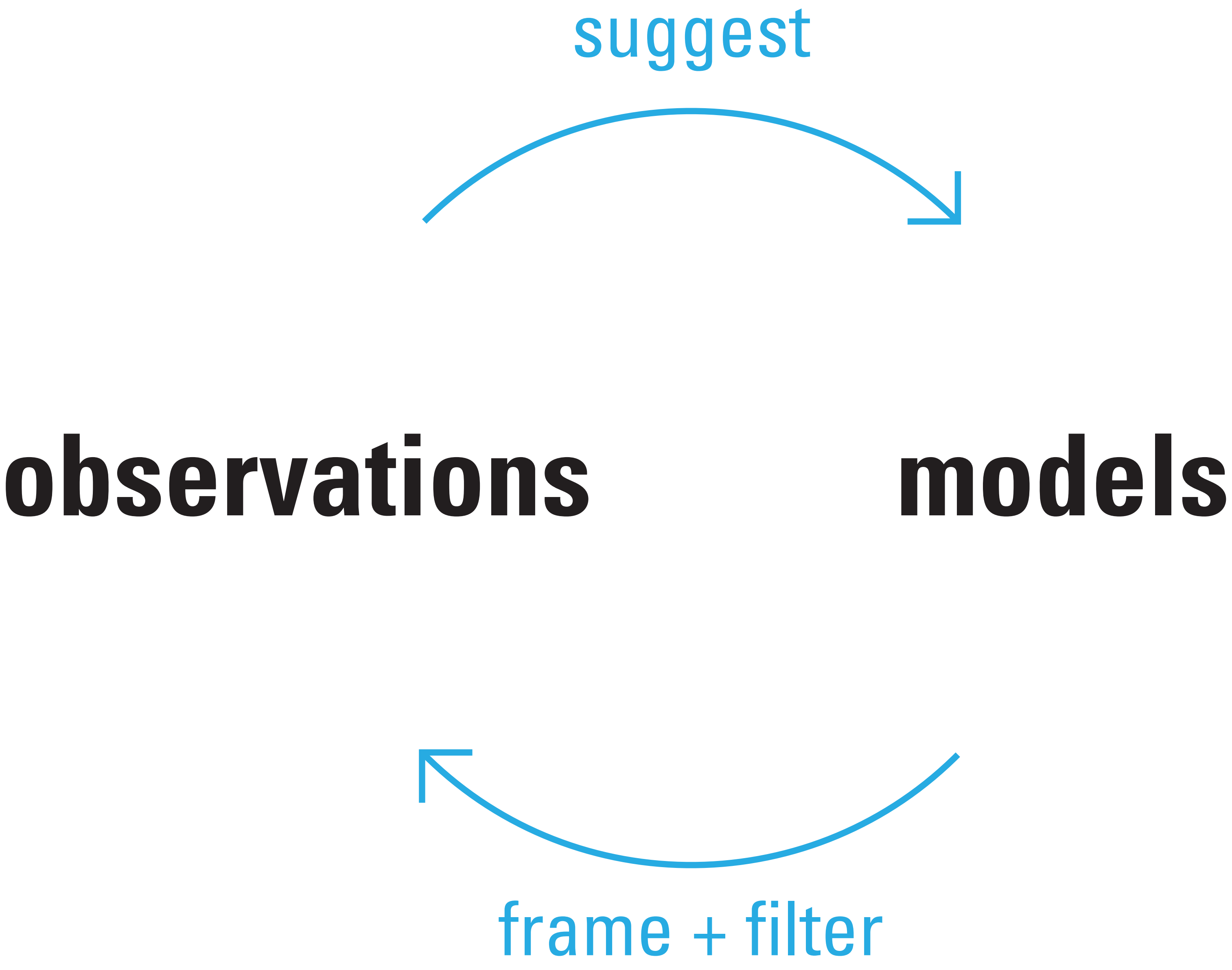
Sharing a model may also change it.



All of these feedback loops, and more, act simultaneously—shaping and reshaping our models.



What we see affects our models, and our models affect what we see.

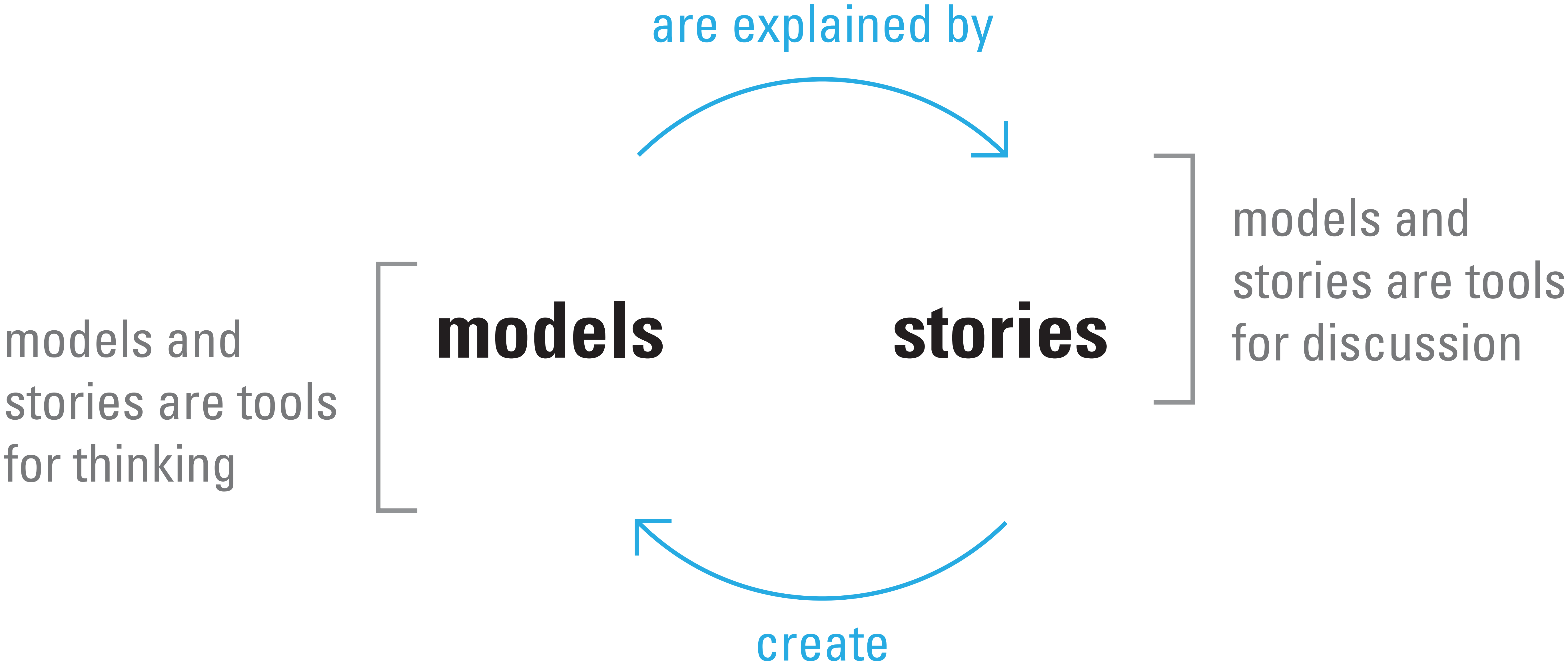

suggest

observations

models

frame + filter

Models help you tell stories, and stories form models in your mind.



are explained by

models and
stories are tools
for thinking

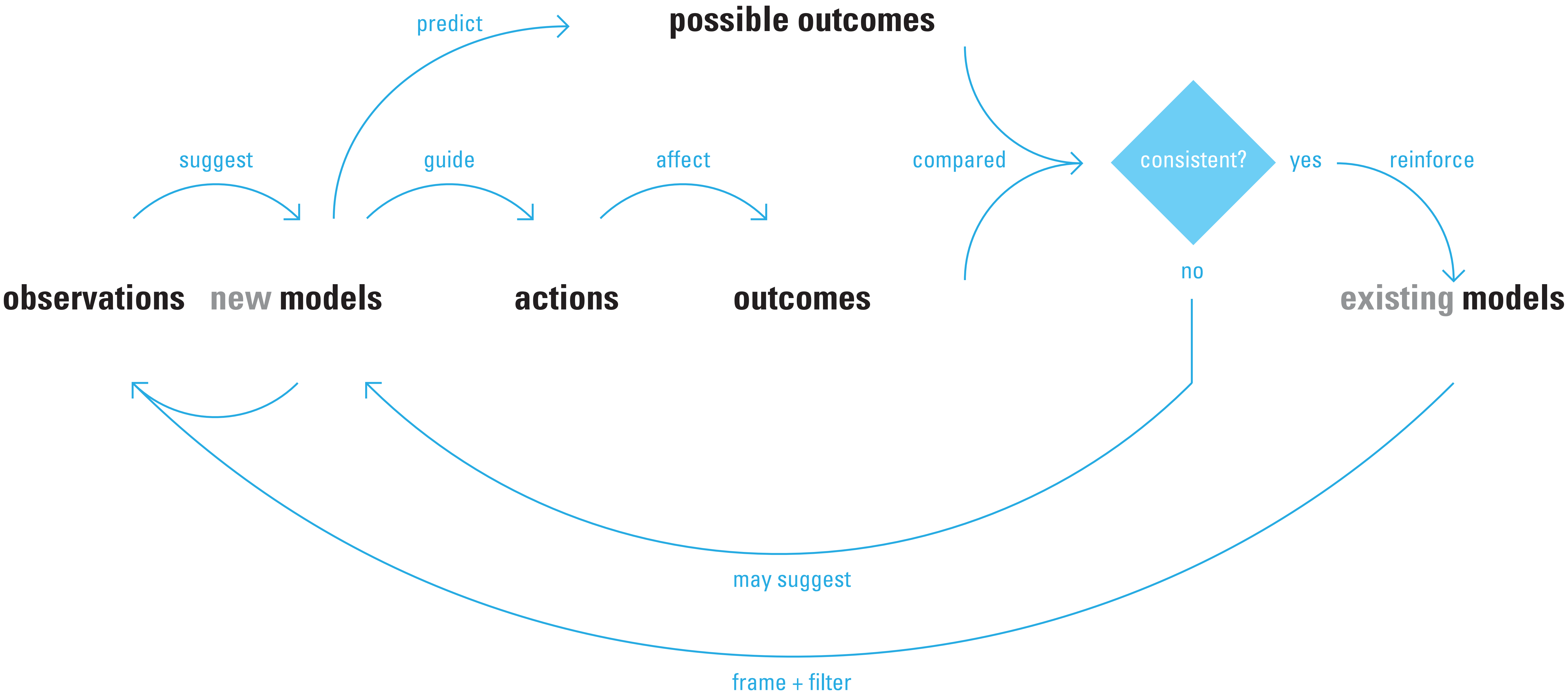
models

stories

models and
stories are tools
for discussion

create

We evolve our models as we test their predictions.



predict

possible outcomes

suggest

guide

affect

compared

consistent?

yes

reinforce

no

observations new **models**

actions

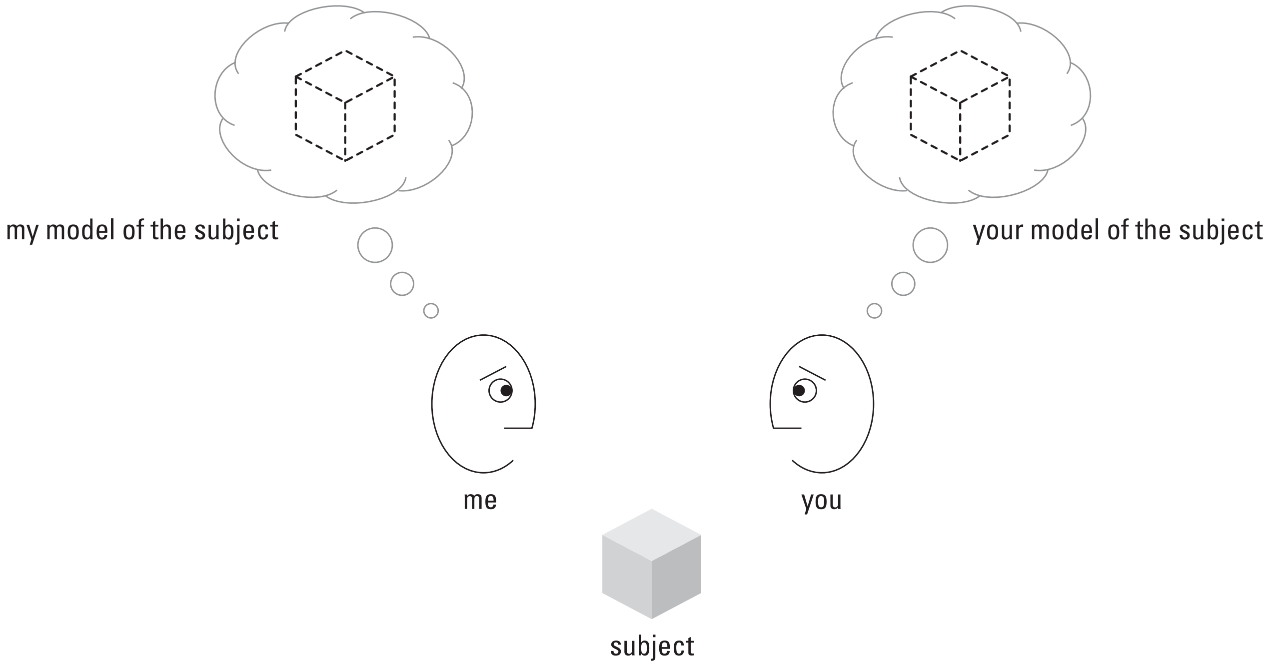
outcomes

existing **models**

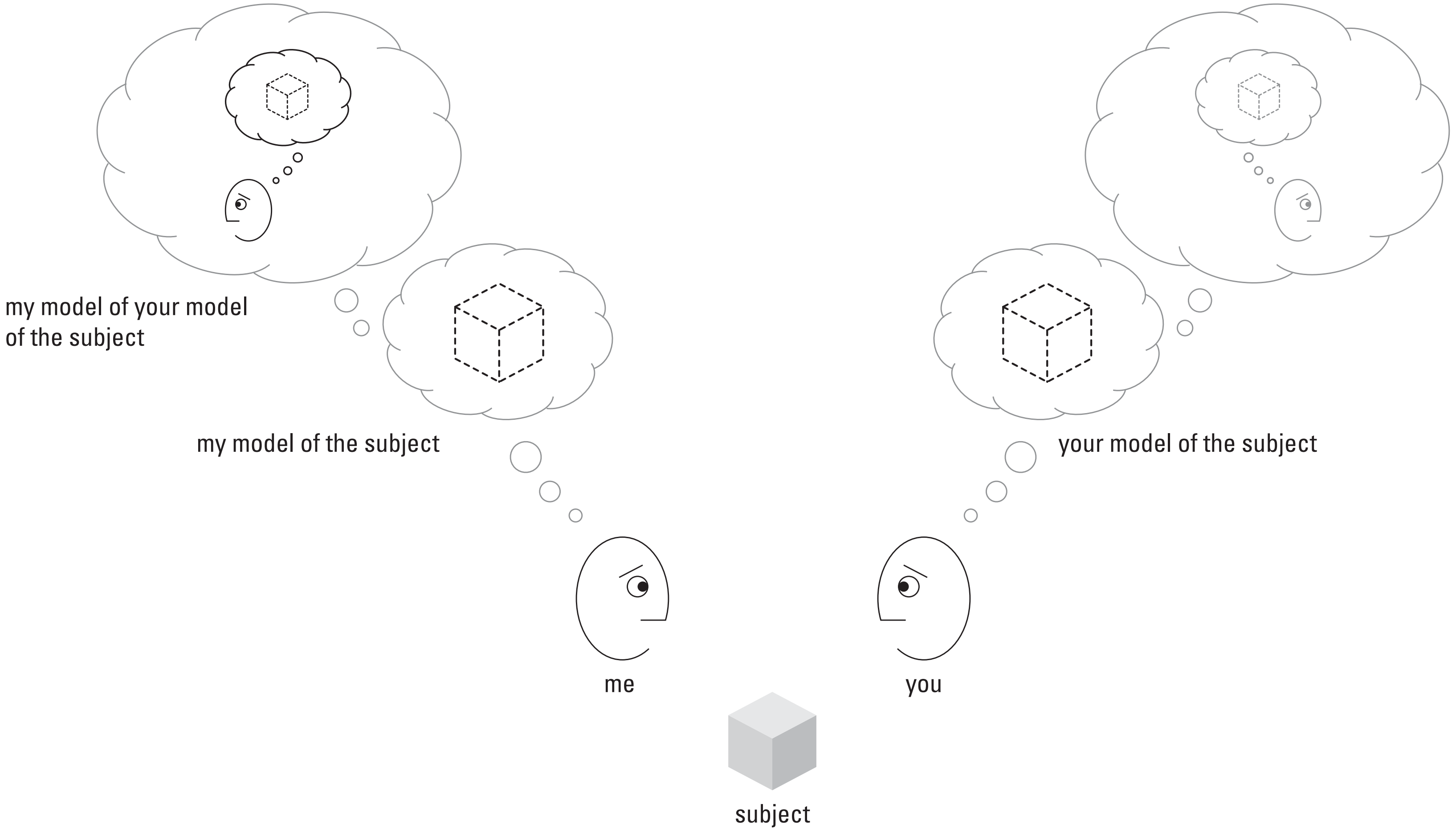
may suggest

frame + filter

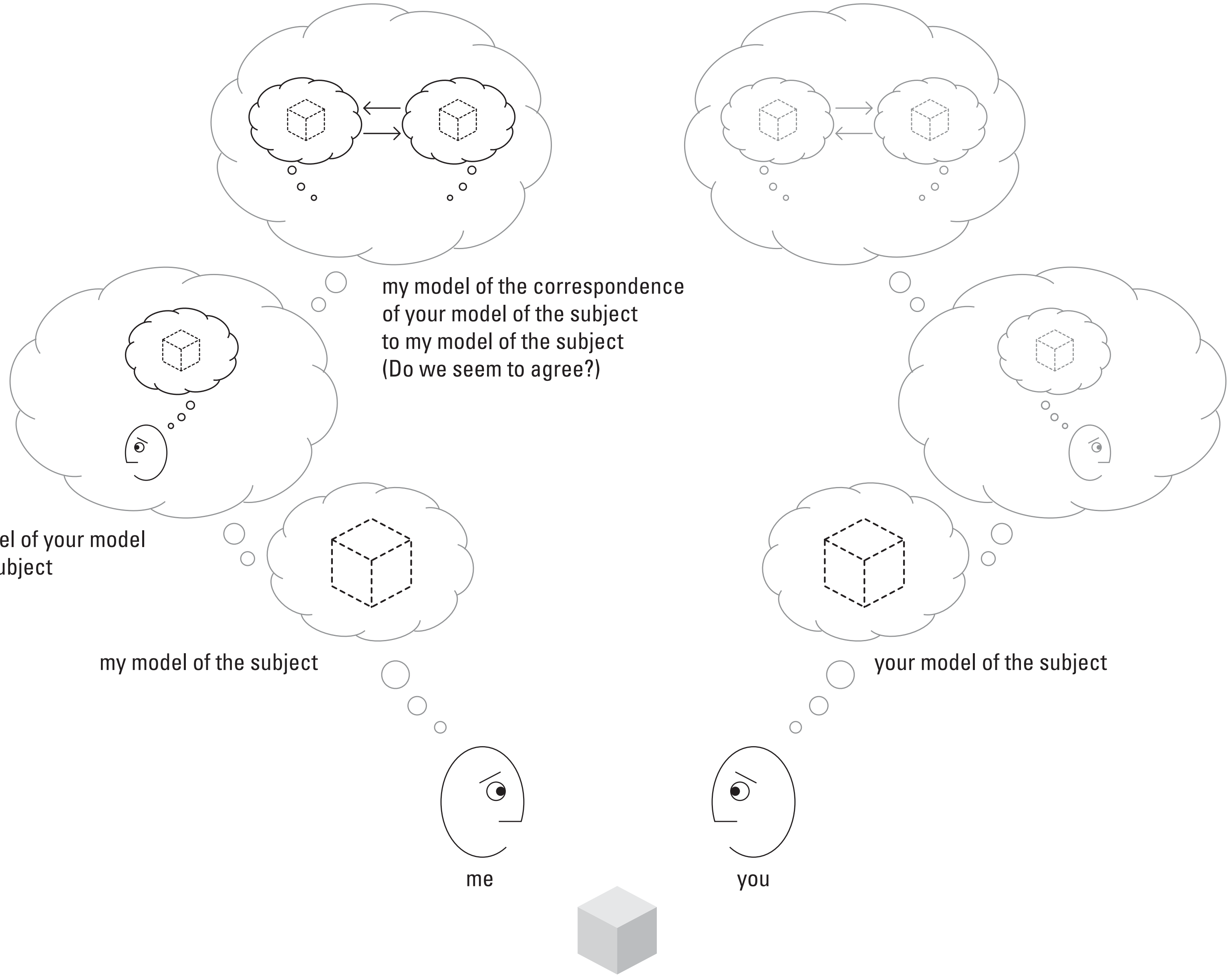
Shared models are the basis for understanding, agreement, and action.



Shared models are the basis for understanding, agreement, and action.



Shared models are the basis for understanding, agreement, and action.



my model of the correspondence
of your model of the subject
to my model of the subject
(Do we seem to agree?)

my model of your model
of the subject

my model of the subject

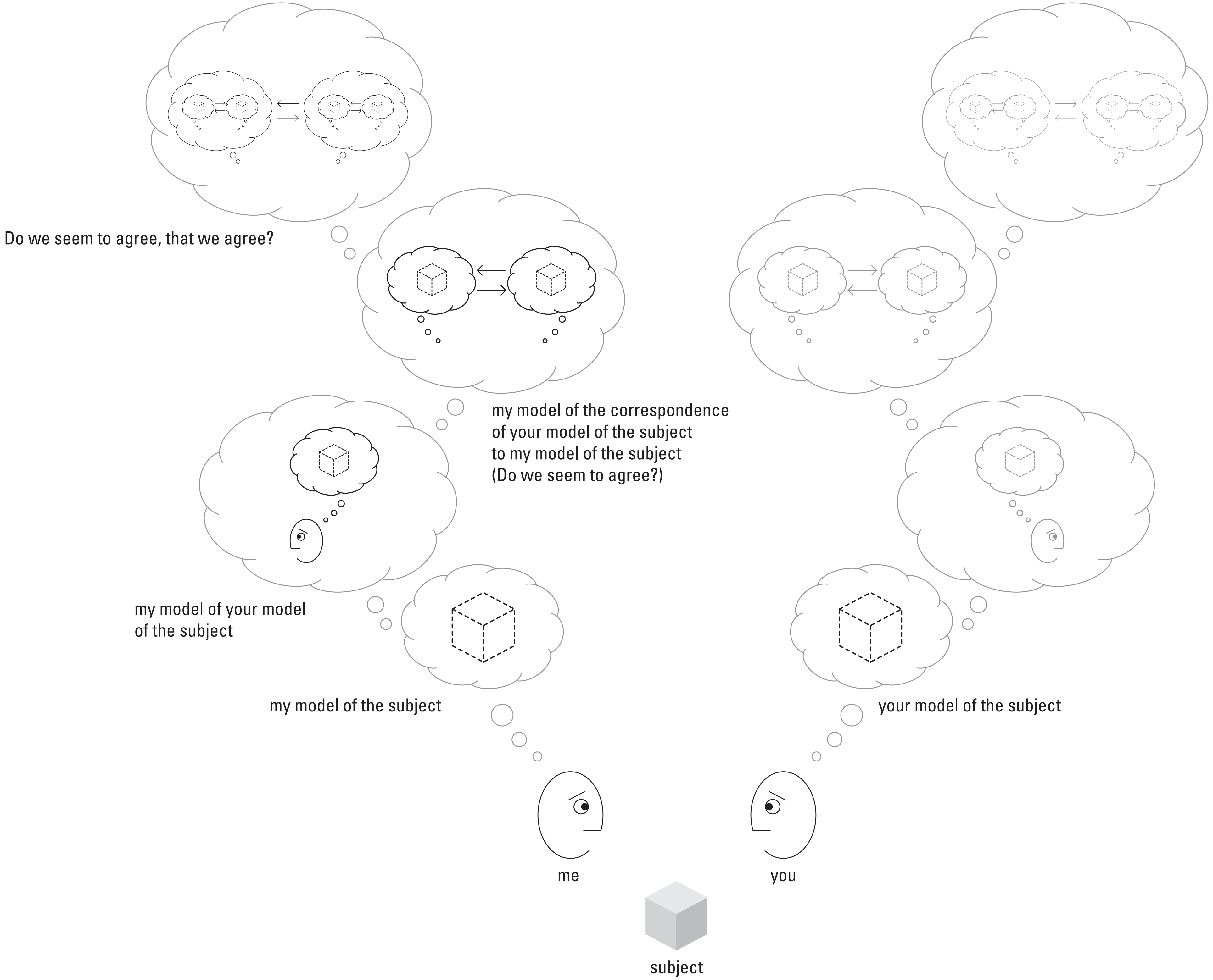
your model of the subject

me

you

subject

Shared models are the basis for understanding, agreement, and action.



Do we seem to agree, that we agree?

my model of the correspondence
of your model of the subject
to my model of the subject
(Do we seem to agree?)

my model of your model
of the subject

my model of the subject

your model of the subject

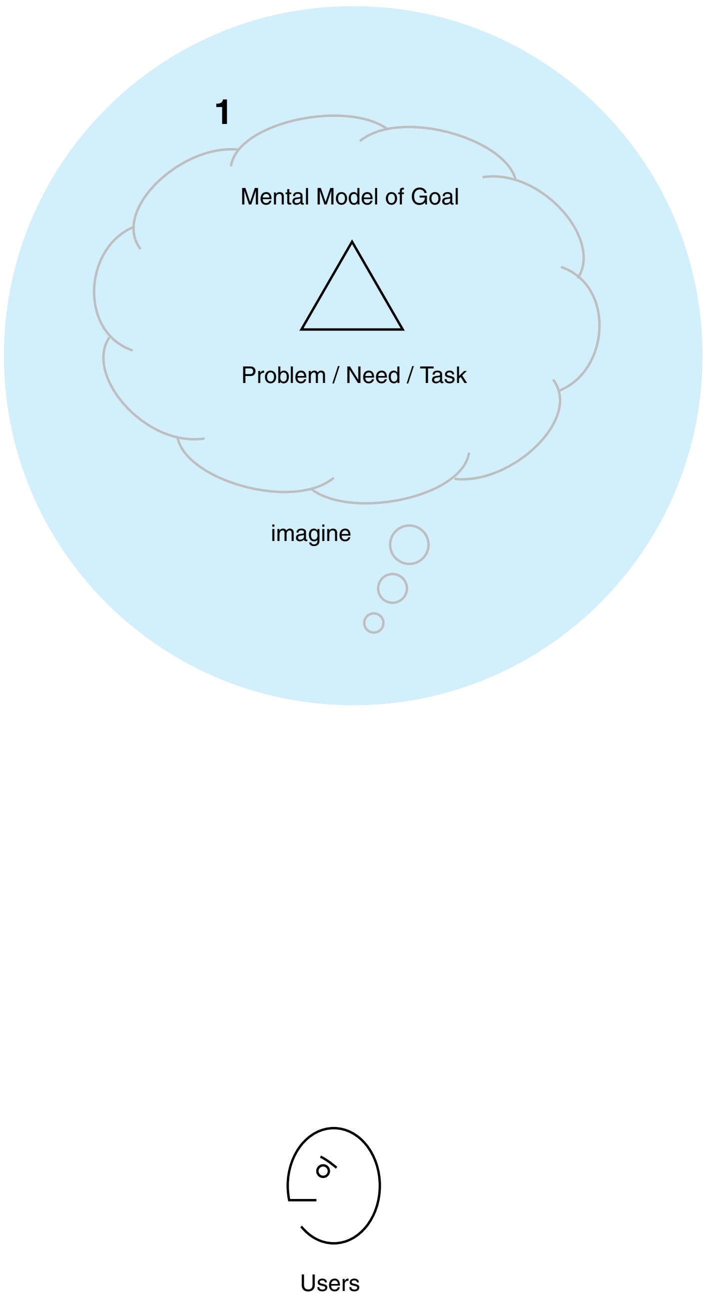
me

you

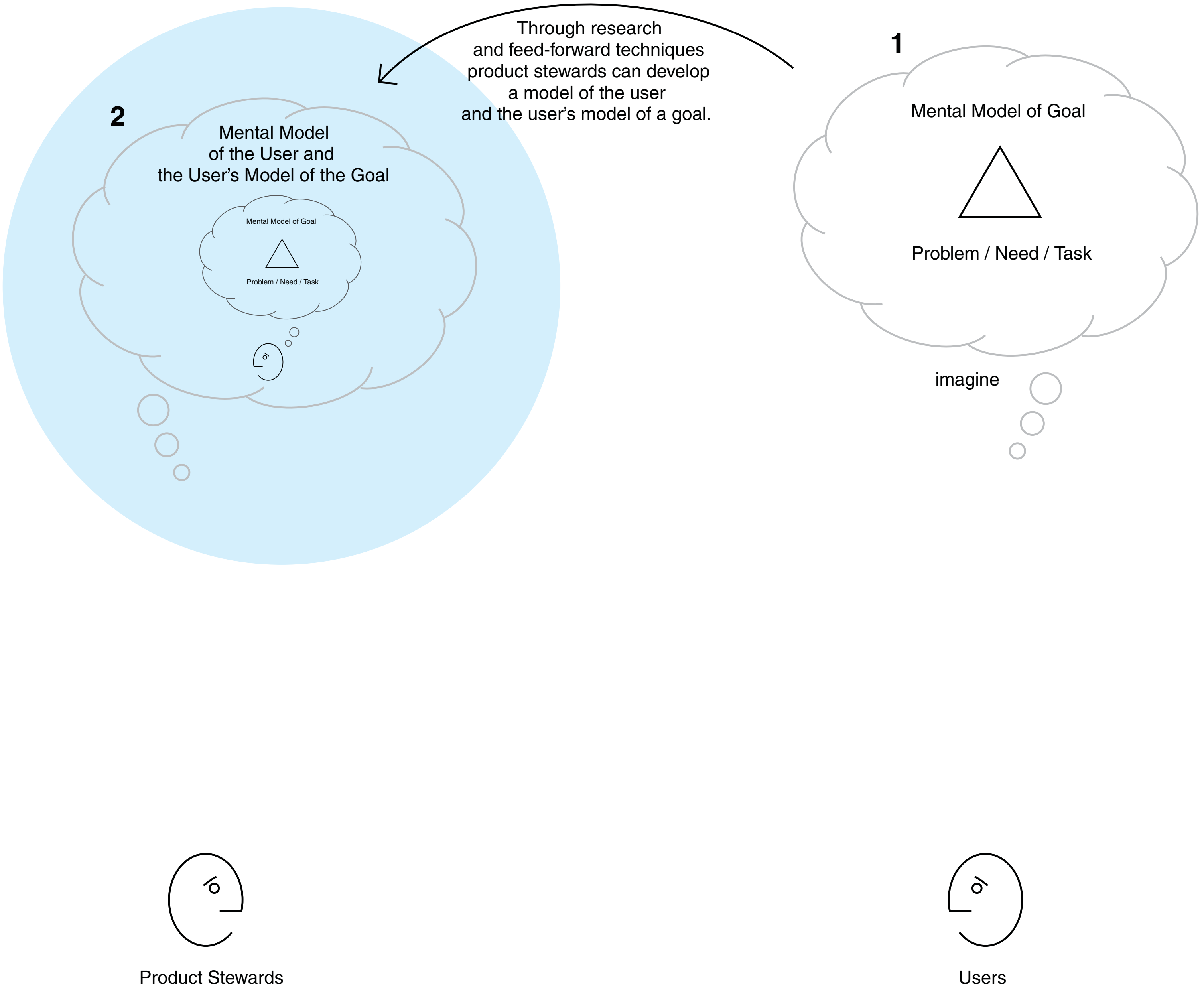
subject

The role of models in product development

1. Users imagine goals

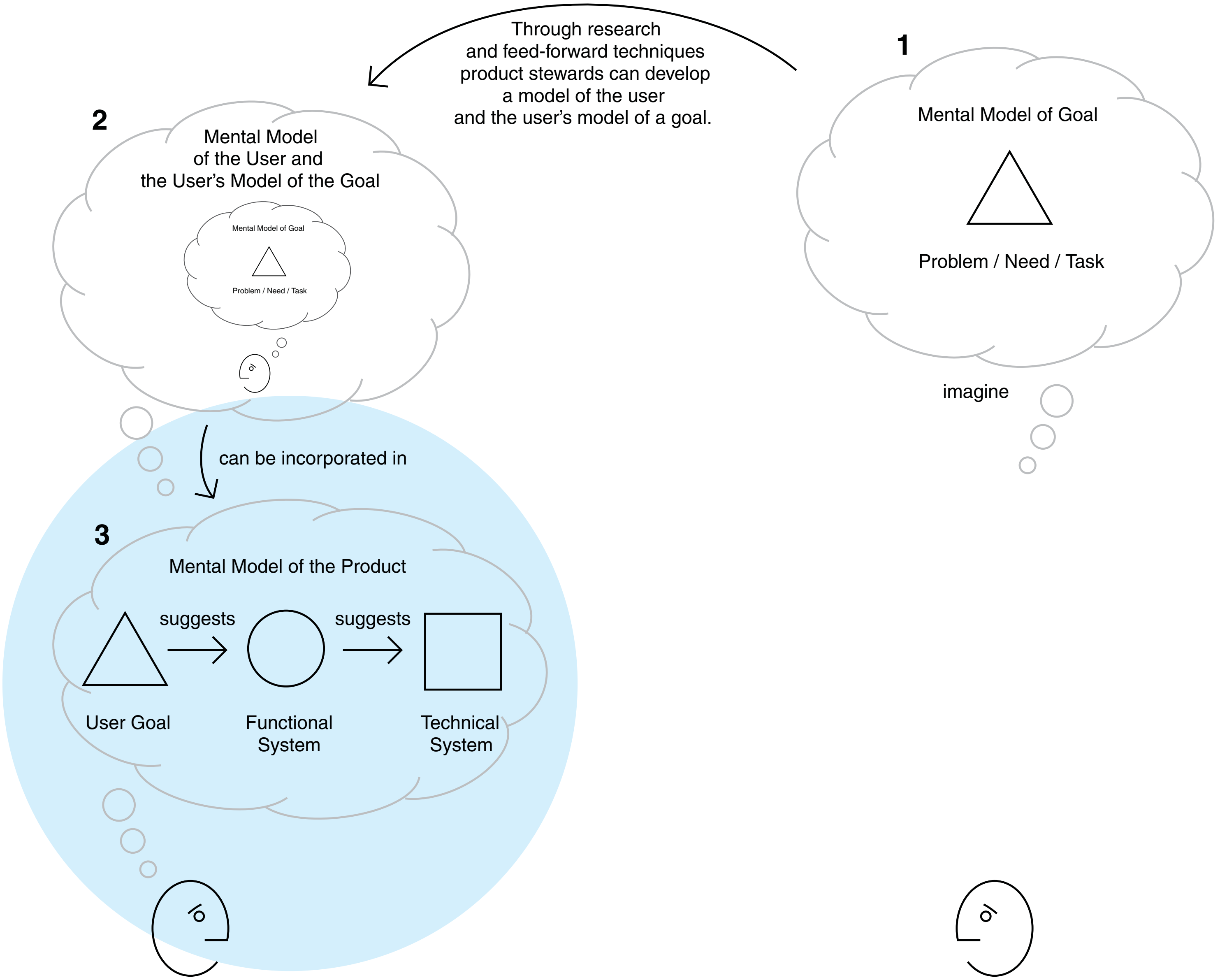


2. Product stewards develop a model of the users



(often through storytelling, story boarding, or scenario planning)

3. Product stewards develop a model of the product



Through research
and feed-forward techniques
product stewards can develop
a model of the user
and the user's model of a goal.

1

Mental Model of Goal

Problem / Need / Task

imagine

2

Mental Model
of the User and
the User's Model of the Goal

Mental Model of Goal

Problem / Need / Task

can be incorporated in

3

Mental Model of the Product

suggests

suggests

User Goal

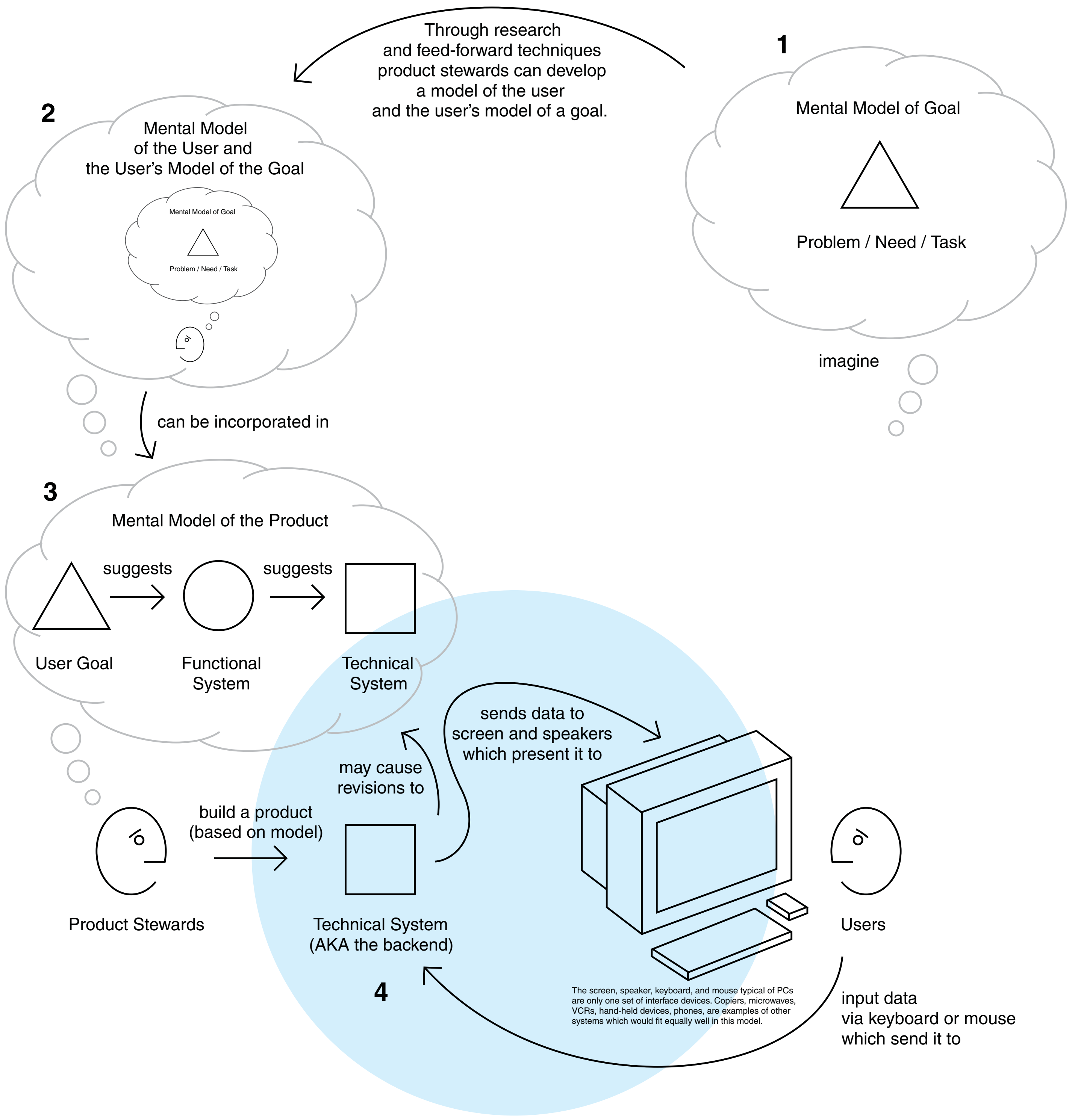
Functional
System

Technical
System

Product Stewards

Users

4. Product stewards deliver a product, and users interact with it



1

Mental Model of Goal

Problem / Need / Task

imagine

Through research
and feed-forward techniques
product stewards can develop
a model of the user
and the user's model of a goal.

2

Mental Model
of the User and
the User's Model of the Goal

Mental Model of Goal

Problem / Need / Task

can be incorporated in

3

Mental Model of the Product

suggests

suggests

User Goal

Functional
System

Technical
System

may cause
revisions to

sends data to
screen and speakers
which present it to

build a product
(based on model)

Product Stewards

Technical System
(AKA the backend)

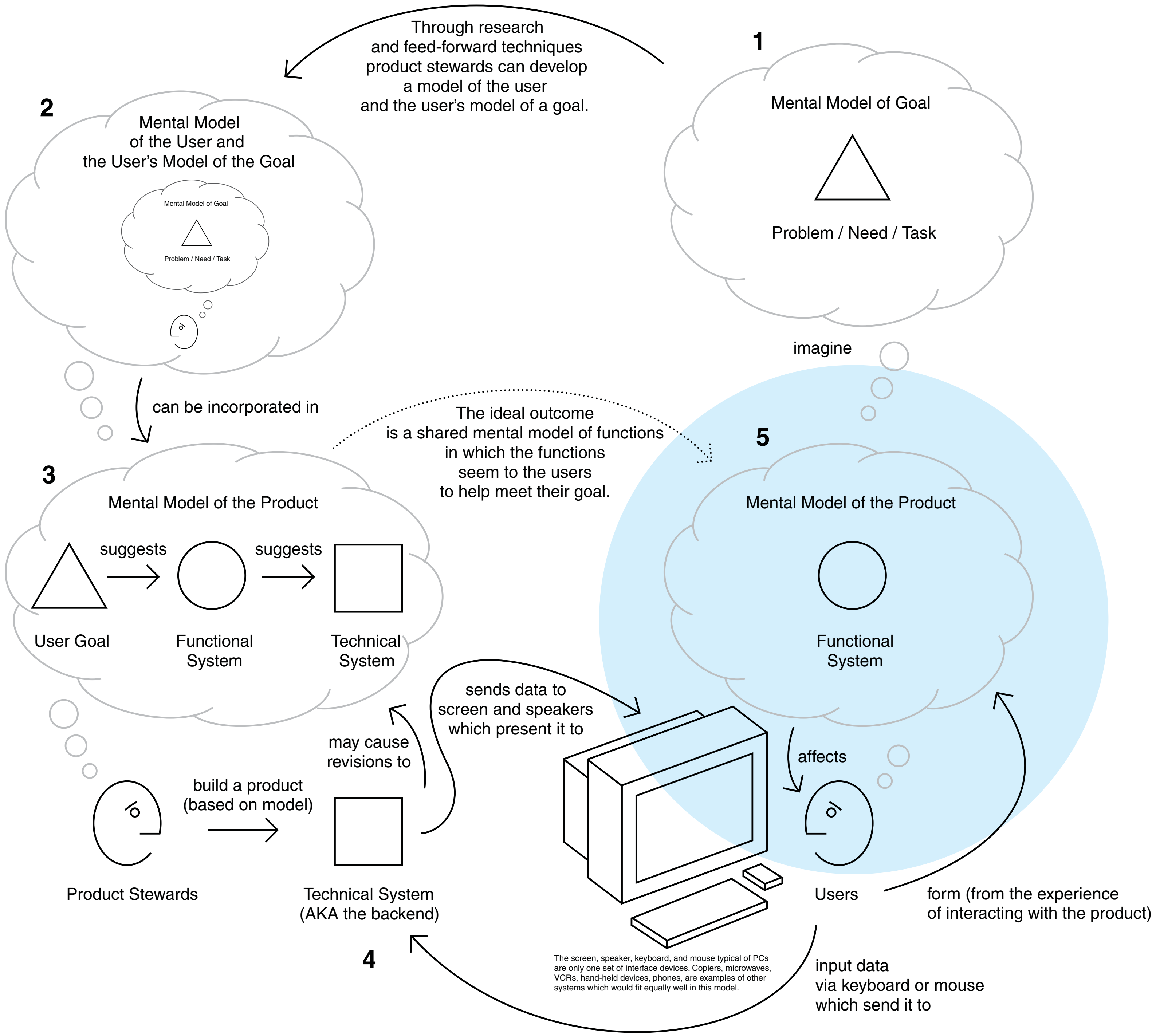
Users

4

The screen, speaker, keyboard, and mouse typical of PCs are only one set of interface devices. Copiers, microwaves, VCRs, hand-held devices, phones, are examples of other systems which would fit equally well in this model.

input data
via keyboard or mouse
which send it to

5. As users interact with the product, they develop a model of how it works



1

Mental Model of Goal

Problem / Need / Task

Through research
and feed-forward techniques
product stewards can develop
a model of the user
and the user's model of a goal.

2

Mental Model
of the User and
the User's Model of the Goal

Mental Model of Goal

Problem / Need / Task

can be incorporated in

3

Mental Model of the Product

suggests

suggests

User Goal

Functional
System

Technical
System

The ideal outcome
is a shared mental model of functions
in which the functions
seem to the users
to help meet their goal.

Product Stewards

build a product
(based on model)

Technical System
(AKA the backend)

may cause
revisions to

sends data to
screen and speakers
which present it to

4

The screen, speaker, keyboard, and mouse typical of PCs are only one set of interface devices. Copiers, microwaves, VCRs, hand-held devices, phones, are examples of other systems which would fit equally well in this model.

input data
via keyboard or mouse
which send it to

imagine

5

Mental Model of the Product

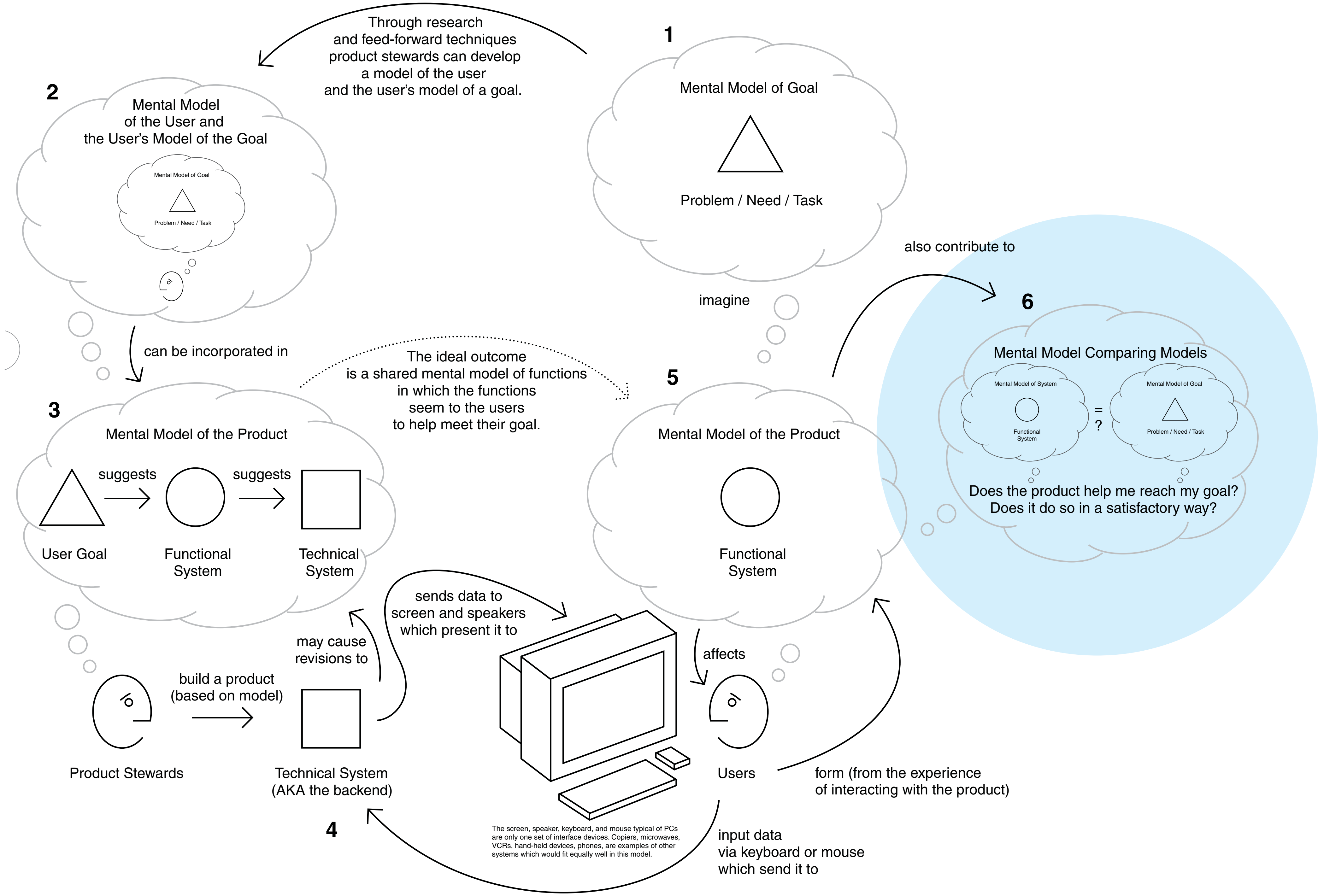
Functional
System

affects

Users

form (from the experience
of interacting with the product)

6. Users compare their model of goals with their model of the product



1

Mental Model of Goal

Problem / Need / Task

Through research
and feed-forward techniques
product stewards can develop
a model of the user
and the user's model of a goal.

2

Mental Model
of the User and
the User's Model of the Goal

Mental Model of Goal

Problem / Need / Task

can be incorporated in

3

Mental Model of the Product

User Goal

suggests

Functional System

suggests

Technical System

The ideal outcome
is a shared mental model of functions
in which the functions
seem to the users
to help meet their goal.

Product Stewards

build a product
(based on model)

Technical System
(AKA the backend)

may cause
revisions to

4

sends data to
screen and speakers
which present it to

The screen, speaker, keyboard, and mouse typical of PCs are only one set of interface devices. Copiers, microwaves, VCRs, hand-held devices, phones, are examples of other systems which would fit equally well in this model.

Users

input data
via keyboard or mouse
which send it to

imagine

5

Mental Model of the Product

Functional
System

affects

form (from the experience
of interacting with the product)

also contribute to

6

Mental Model Comparing Models

Mental Model of System

Functional
System

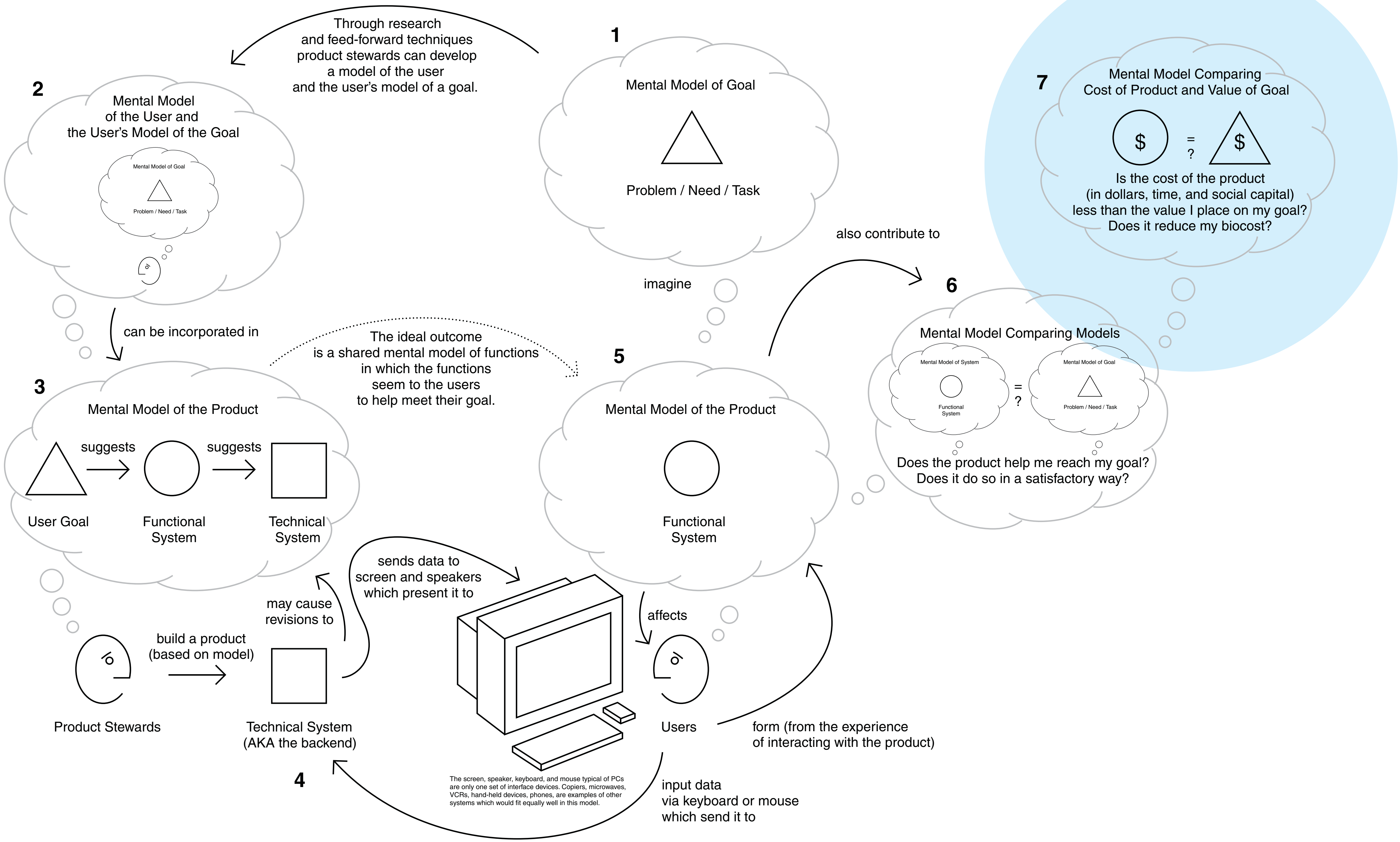
=
?

Mental Model of Goal

Problem / Need / Task

Does the product help me reach my goal?
Does it do so in a satisfactory way?

7. Users also compare cost and value



Through research
and feed-forward techniques
product stewards can develop
a model of the user
and the user's model of a goal.

1

Mental Model of Goal

Problem / Need / Task

imagine

also contribute to

2

Mental Model
of the User and
the User's Model of the Goal

Mental Model of Goal

Problem / Need / Task

can be incorporated in

3

Mental Model of the Product

suggests

suggests

User Goal

Functional
System

Technical
System

The ideal outcome
is a shared mental model of functions
in which the functions
seem to the users
to help meet their goal.

5

Mental Model of the Product

Functional
System

6

Mental Model Comparing Models

Mental Model of System

Functional
System

=
?

Mental Model of Goal

Problem / Need / Task

Does the product help me reach my goal?
Does it do so in a satisfactory way?

7

Mental Model Comparing
Cost of Product and Value of Goal

$

=
?

$

Is the cost of the product
(in dollars, time, and social capital)
less than the value I place on my goal?
Does it reduce my biocost?

Product Stewards

build a product
(based on model)

may cause
revisions to

Technical System
(AKA the backend)

4

sends data to
screen and speakers
which present it to

The screen, speaker, keyboard, and mouse typical of PCs are only one set of interface devices. Copiers, microwaves, VCRs, hand-held devices, phones, are examples of other systems which would fit equally well in this model.

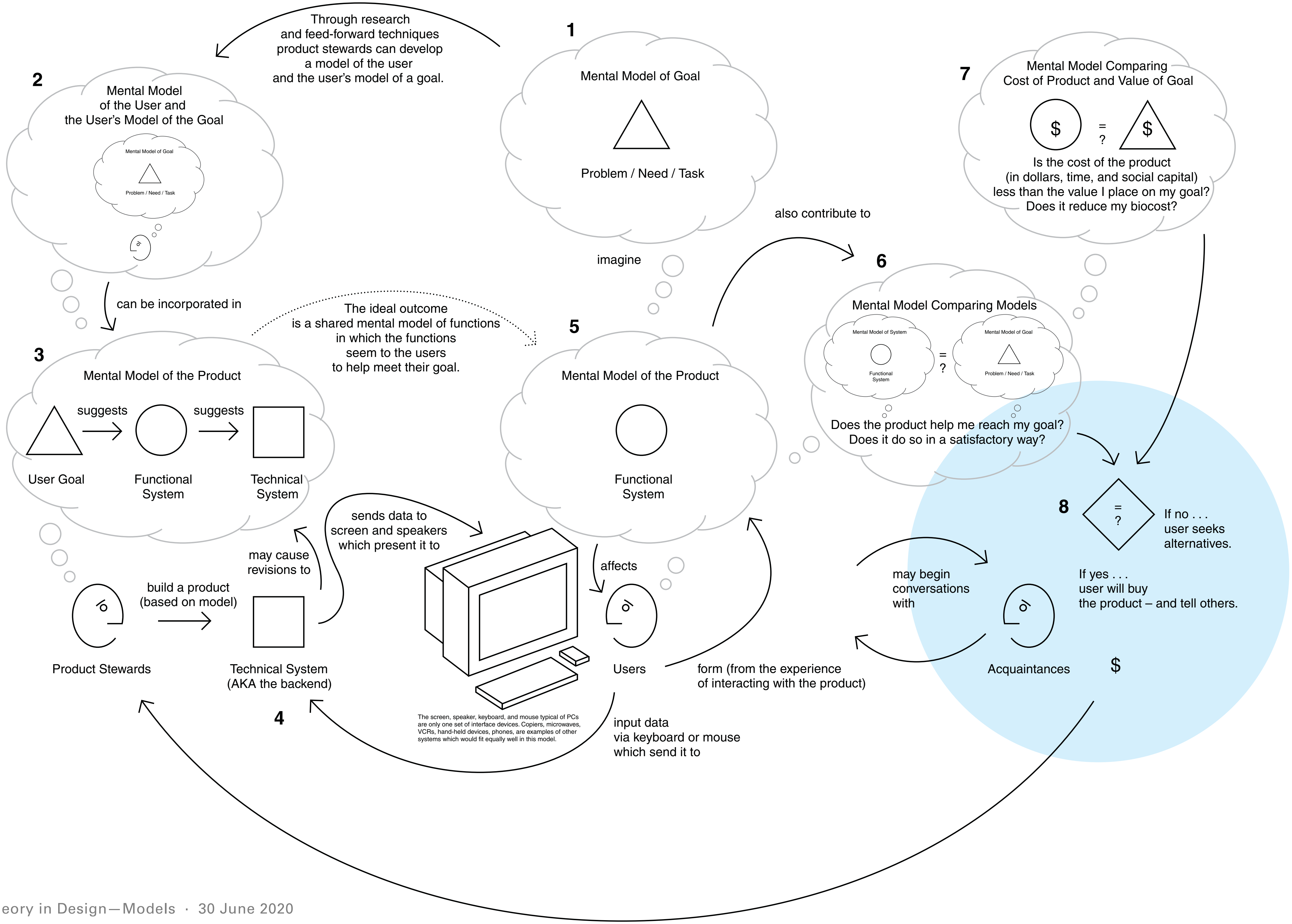
affects

Users

form (from the experience
of interacting with the product)

input data
via keyboard or mouse
which send it to

8. If cost < value, users may buy the product and tell others about it



1

Mental Model of Goal

Problem / Need / Task

Through research
and feed-forward techniques
product stewards can develop
a model of the user
and the user's model of a goal.

2

Mental Model
of the User and
the User's Model of the Goal

can be incorporated in

3

Mental Model of the Product

User Goal

suggests

Functional
System

suggests

Technical
System

The ideal outcome
is a shared mental model of functions
in which the functions
seem to the users
to help meet their goal.

Product Stewards

build a product
(based on model)

Technical System
(AKA the backend)

may cause
revisions to

4

sends data to
screen and speakers
which present it to

The screen, speaker, keyboard, and mouse typical of PCs are only one set of interface devices. Copiers, microwaves, VCRs, hand-held devices, phones, are examples of other systems which would fit equally well in this model.

imagine

5

Mental Model of the Product

Functional
System

affects

Users

input data
via keyboard or mouse
which send it to

form (from the experience
of interacting with the product)

also contribute to

6

Mental Model Comparing Models

Does the product help me reach my goal?
Does it do so in a satisfactory way?

7

Mental Model Comparing
Cost of Product and Value of Goal

Is the cost of the product
(in dollars, time, and social capital)
less than the value I place on my goal?
Does it reduce my biocost?

8

If no . . .
user seeks
alternatives.

If yes . . .
user will buy
the product – and tell others.

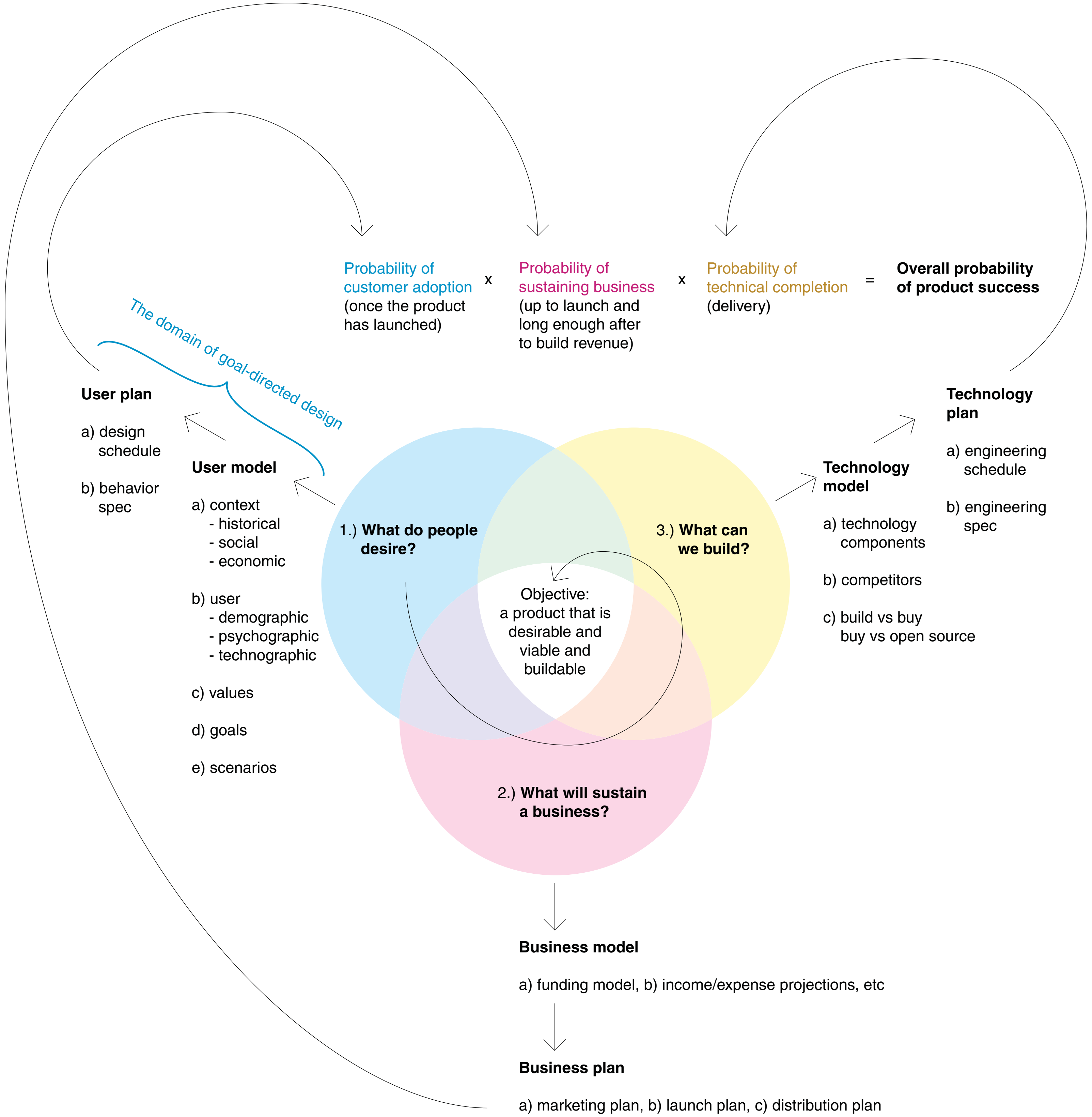
may begin
conversations
with

Acquaintances

The balanced innovation model

Successful products balance three forces

- People’s needs
- Business costs
- Technology limits



Probability of customer adoption (once the product has launched) x Probability of sustaining business (up to launch and long enough after to build revenue) x Probability of technical completion (delivery) = **Overall probability of product success**

The domain of goal-directed design

User plan

a) design schedule

b) behavior spec

User model

a) context
  - historical
  - social
  - economic

b) user
  - demographic
  - psychographic
  - technographic

c) values

d) goals

e) scenarios

1.) **What do people desire?**

3.) **What can we build?**

Objective: a product that is desirable and viable and buildable

2.) **What will sustain a business?**

Technology model

a) technology components

b) competitors

c) build vs buy buy vs open source

Technology plan

a) engineering schedule

b) engineering spec

Business model

a) funding model, b) income/expense projections, etc

Business plan

a) marketing plan, b) launch plan, c) distribution plan

Business-related models

- What is the structure of the business?
- Where does the money come from?
- How does the business add value?
- Where does the money go?
- When does income exceed expense?
- Who are the competitors?

Technology-related models

- What is the structure of the data?
- What is the primary data type?
- Where does the data come from?
- How is the data transformed?
- Where does the data go?

User-related models

- What are the main types of users?
- What are their goals?
- What tasks will they complete to achieve those goals?
- Or—what is the customer journey?
- What is the context of use?

Content or domain-related models

- What is the basic content element or “chunk”?
- What are all of the elements?
- How do they relate to each other?
- Or—what is the structure of the content?

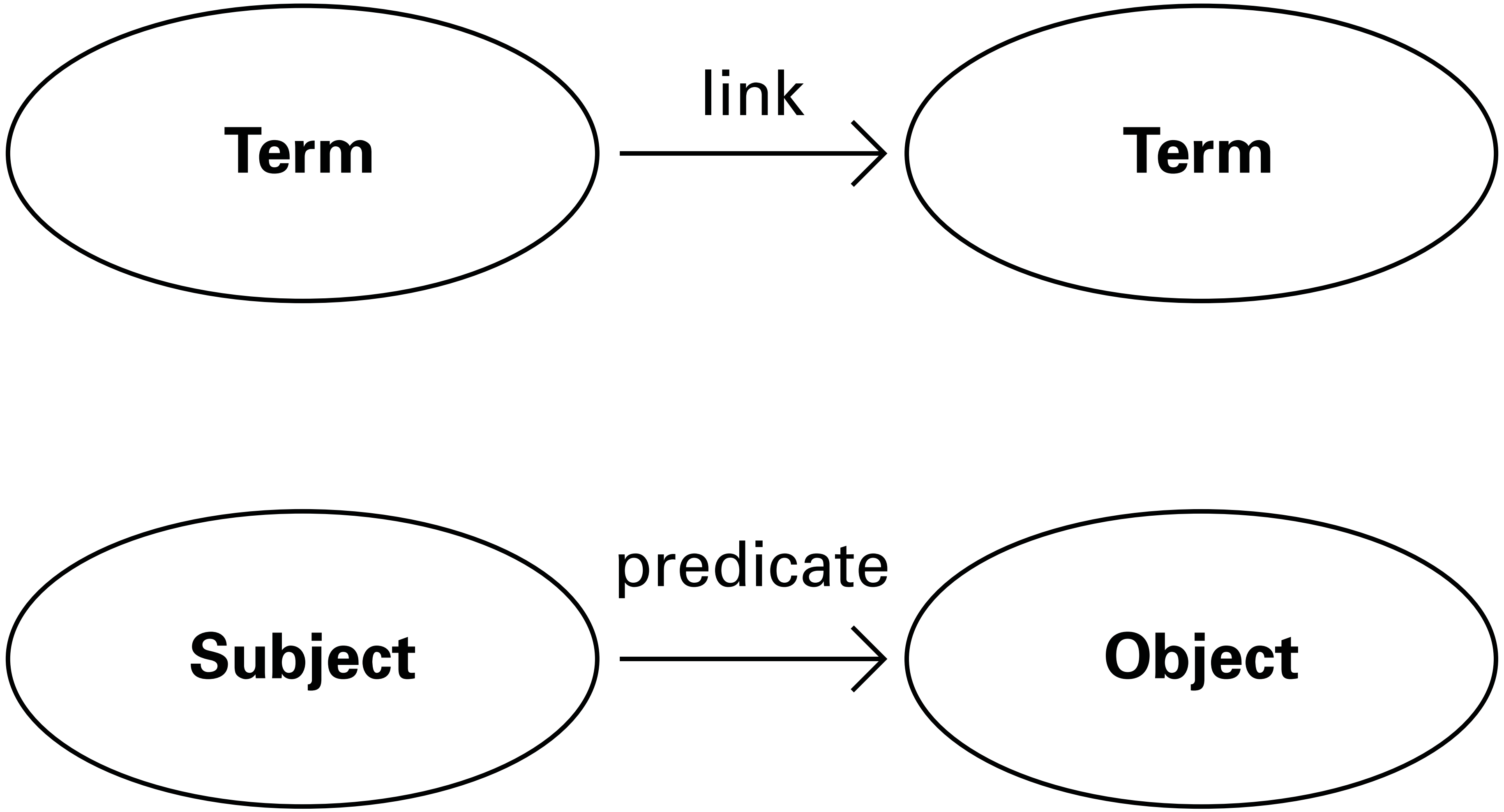
Concept maps — 2 case studies

Concept mapping is a method of making models

Concept maps...

- contain terms—
  related ideas
- show how terms are linked—
  expose connections
- describe the links—
  identify relationships

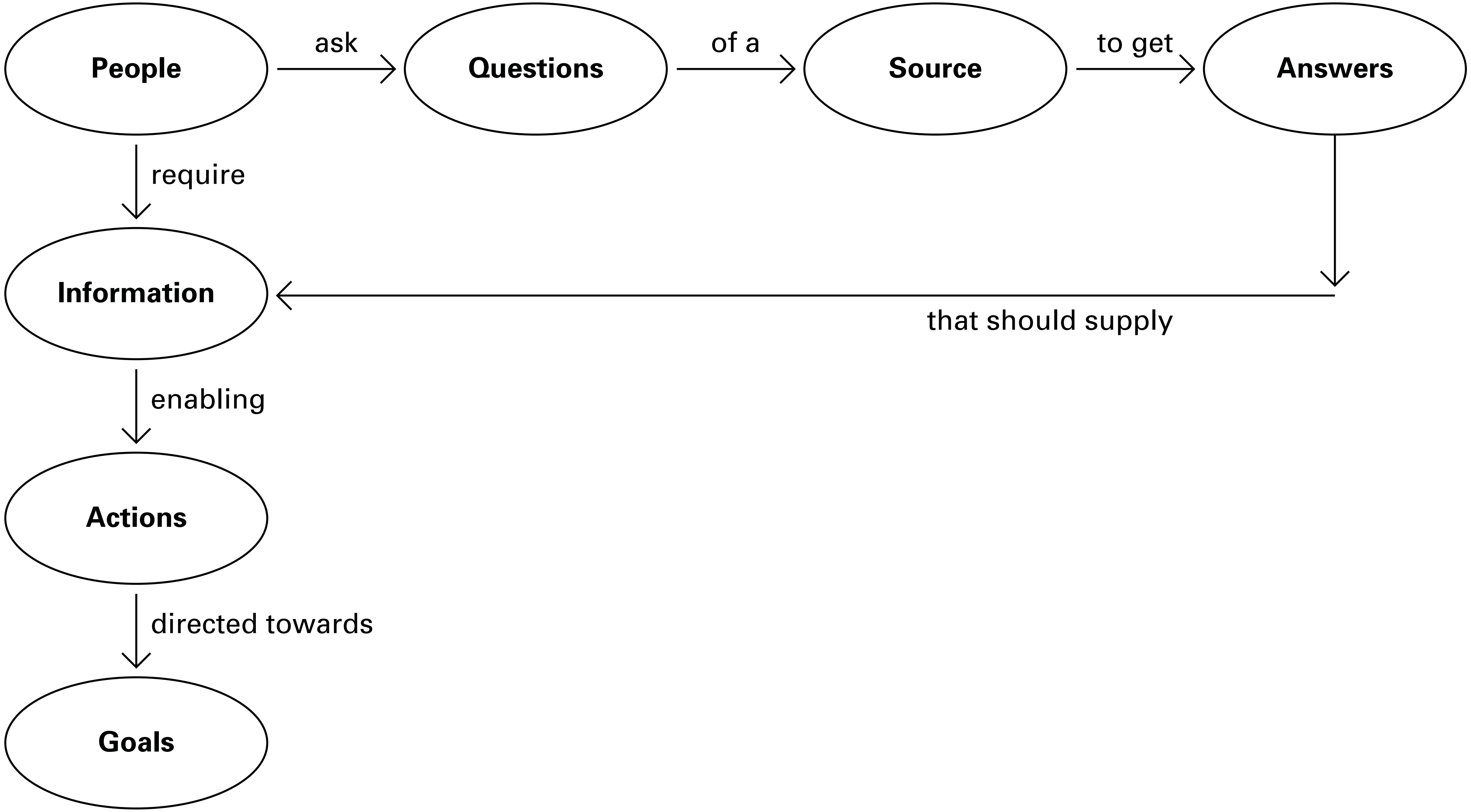
The basic form of concept maps: Verbs link terms to form propositions

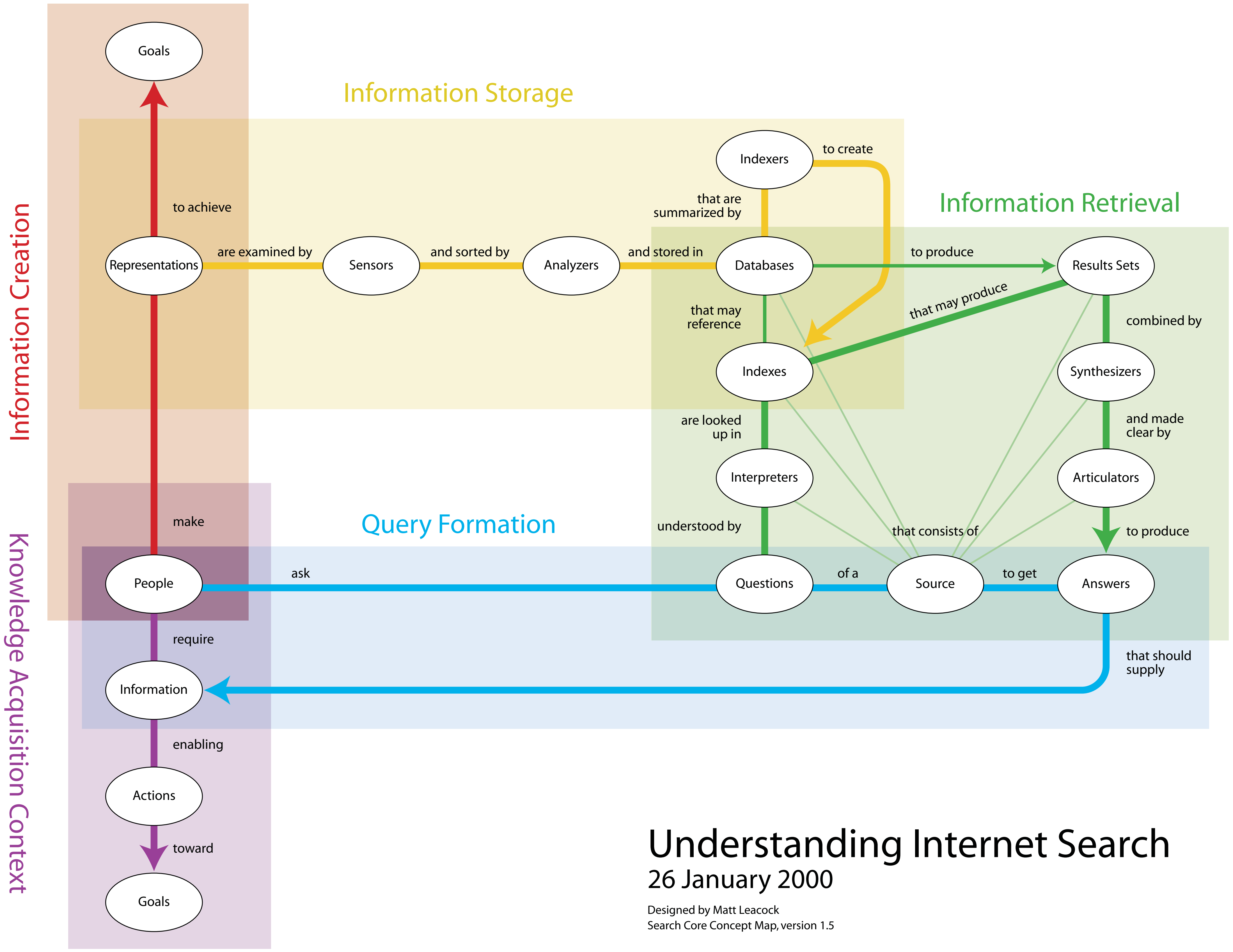


Term link **Term**

Subject predicate **Object**

Example of a simple concept map

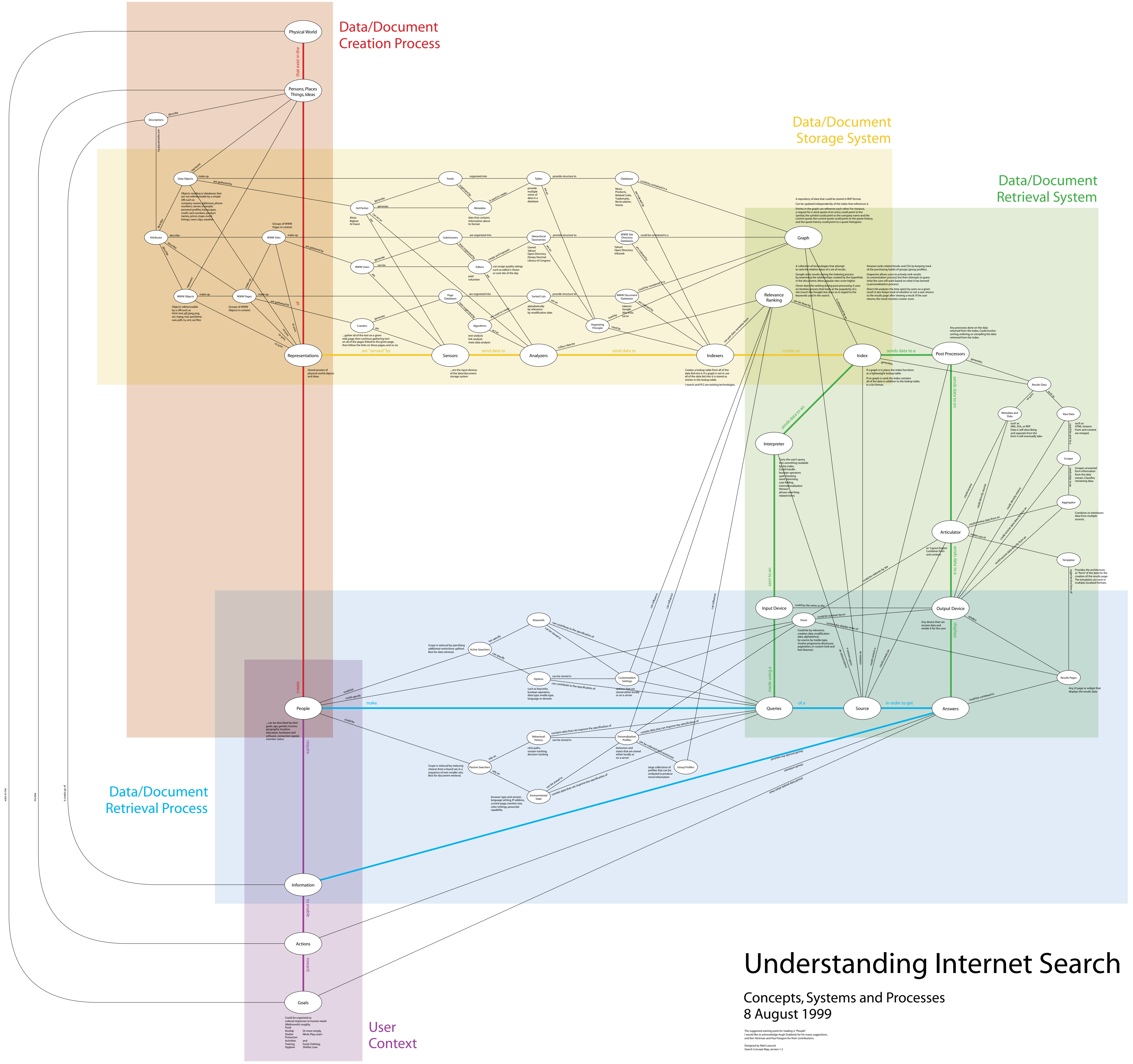




Understanding Internet Search

26 January 2000

Designed by Matt Leacock
Search Core Concept Map, version 1.5



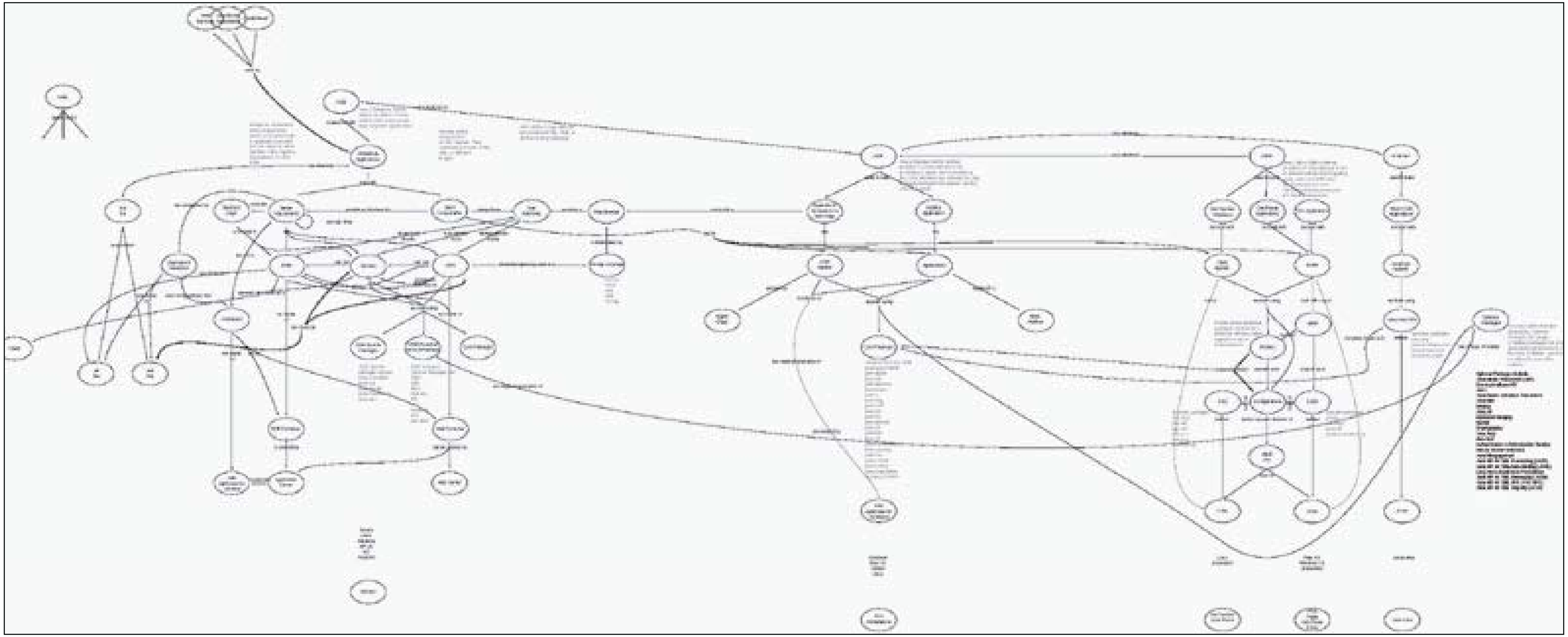
Understanding Internet Search

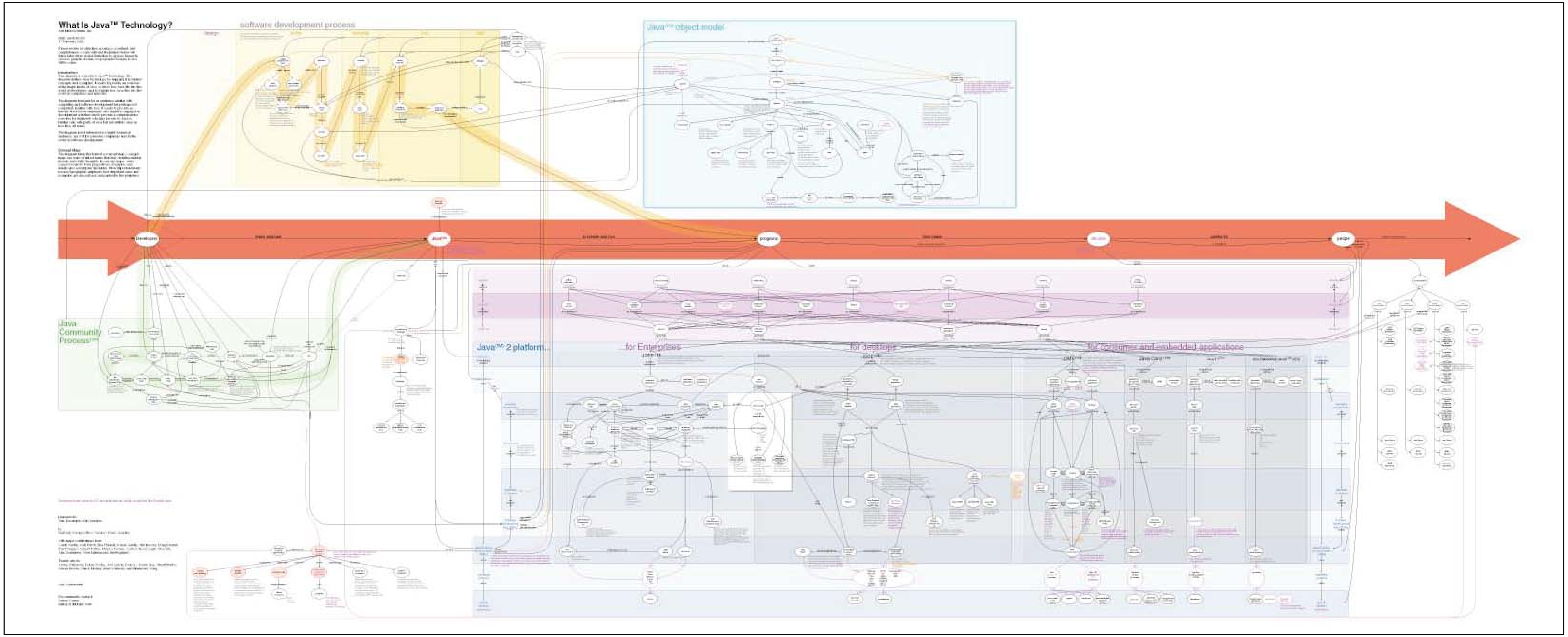
Concepts, Systems and Processes
8 August 1999

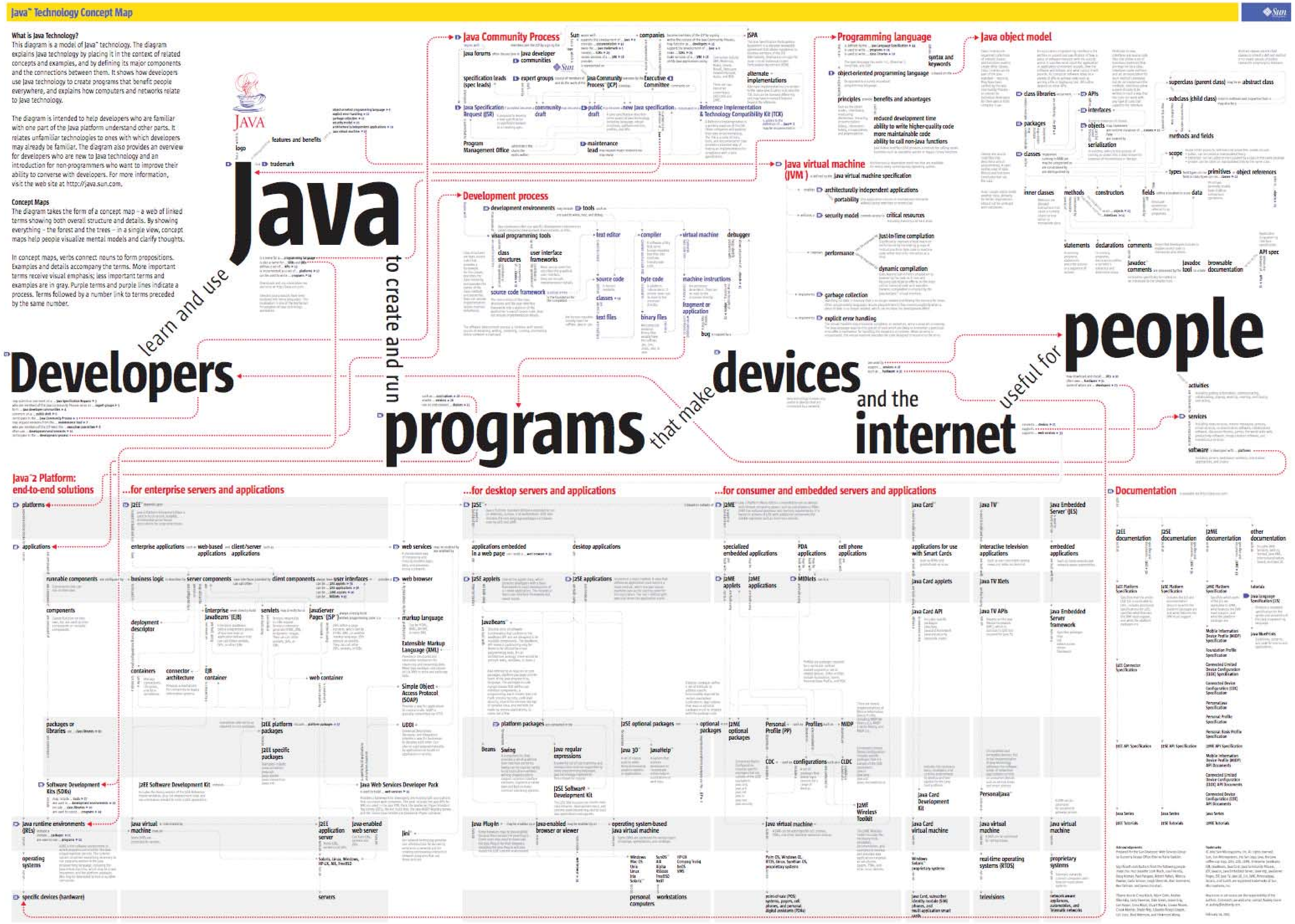
The suggested starting point for reading is "People".
I would like to acknowledge Hugh Dubberly for his many suggestions,
and Ken Hickman and Paul Pangaro for their contributions.

Designed by Matt Leacock
Search Concept Map, version 1.2


Collection
instance
Container
call(function)
deprecation
age concept
methods
instance variable
ment
Collectio
API
Interrupt
Abstract Method
instance
data stru
63




What Is Java™ Technology?
software development process
Java™ object model
developers
Java™
programs
devices
people
Java Community Process℠
Java™ 2 platform...
...for Enterprises
for desktops
for consumer and embedded applications
Java Card™



Java™ Technology Concept Map

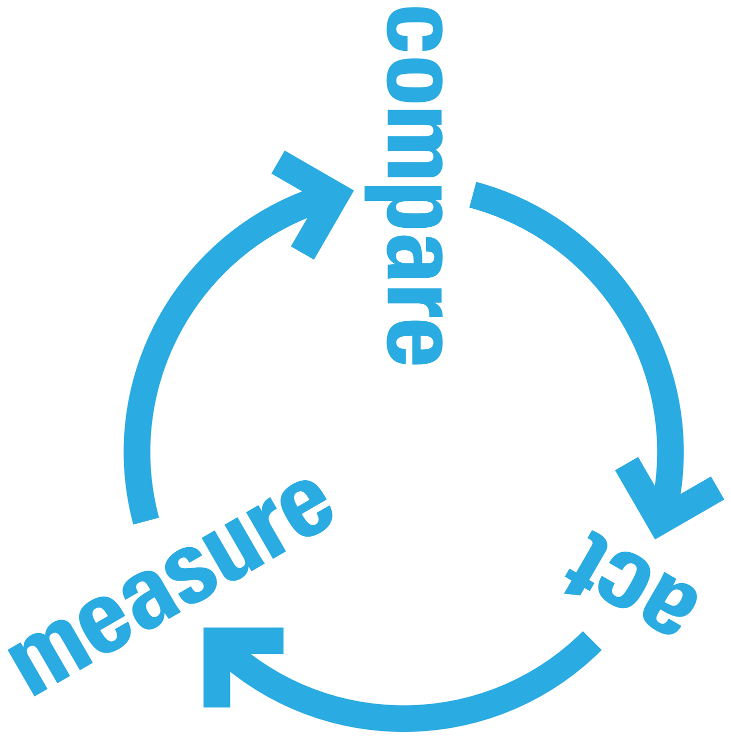
What is Java Technology?
This diagram is a model of Java™ technology. The diagram explains Java technology by placing it in the context of related concepts and examples, and by defining its major components and the connections between them. It shows how developers use Java technology to create programs that benefit people everywhere, and explains how computers and networks relate to Java technology.

The diagram is intended to help developers who are familiar with one part of the Java platform understand other parts. It relates unfamiliar technologies to ones with which developers may already be familiar. The diagram also provides an overview for developers who are new to Java technology and an introduction for non-programmers who want to improve their ability to converse with developers. For more information, visit the web site at http://java.sun.com.

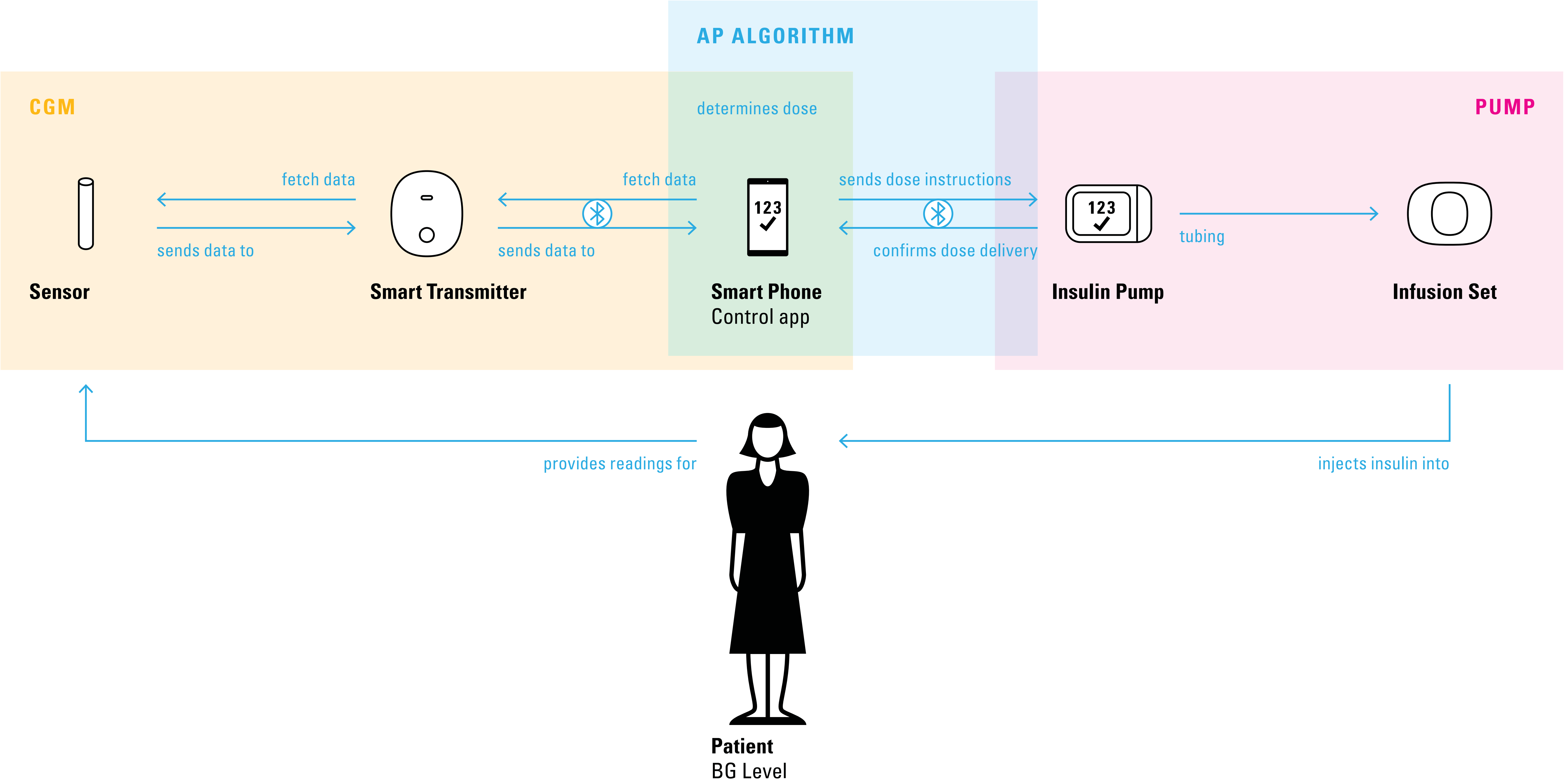
Concept Maps
The diagram takes the form of a concept map – a web of linked terms showing both overall structure and details. By showing everything – the forest and the trees – in a single view, concept maps help people visualize mental models and clarify thoughts.

In concept maps, verbs connect nouns to form propositions. Examples and details accompany the terms. More important terms receive visual emphasis; less important terms and examples are in gray. Purple terms and purple lines indicate a process. Terms followed by a number link to terms preceded by the same number.

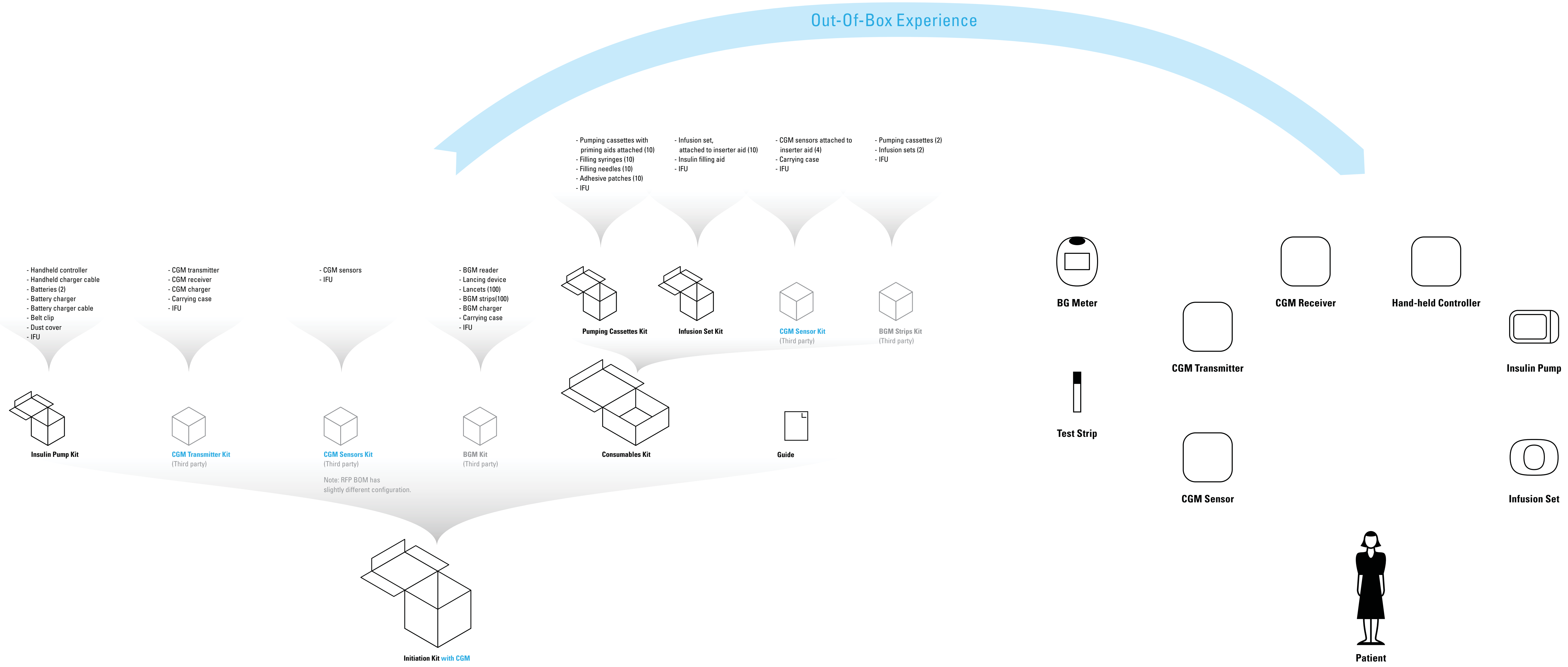
Increasingly, we are designing for complex adaptive systems—systems which involve feedback, learning, and conversation.



Blood glucose + insulin management systems for diabetes patients—an “artificial pancreas”.



Product delivery and set-up systems—the “out-of-box-experience”.



Out-Of-Box Experience

- Pumping cassettes with priming aids attached (10)
- Filling syringes (10)
- Filling needles (10)
- Adhesive patches (10)
- IFU

- Infusion set, attached to inserter aid (10)
- Insulin filling aid
- IFU

- CGM sensors attached to inserter aid (4)
- Carrying case
- IFU

- Pumping cassettes (2)
- Infusion sets (2)
- IFU

Pumping Cassettes Kit

Infusion Set Kit

CGM Sensor Kit
(Third party)

BGM Strips Kit
(Third party)

- Handheld controller
- Handheld charger cable
- Batteries (2)
- Battery charger
- Battery charger cable
- Belt clip
- Dust cover
- IFU

- CGM transmitter
- CGM receiver
- CGM charger
- Carrying case
- IFU

- CGM sensors
- IFU

- BGM reader
- Lancing device
- Lancets (100)
- BGM strips(100)
- BGM charger
- Carrying case
- IFU

Insulin Pump Kit

CGM Transmitter Kit
(Third party)

CGM Sensors Kit
(Third party)

Note: RFP BOM has
slightly different configuration.

BGM Kit
(Third party)

Consumables Kit

Guide

Initiation Kit with CGM

BG Meter

Test Strip

CGM Transmitter

CGM Sensor

CGM Receiver

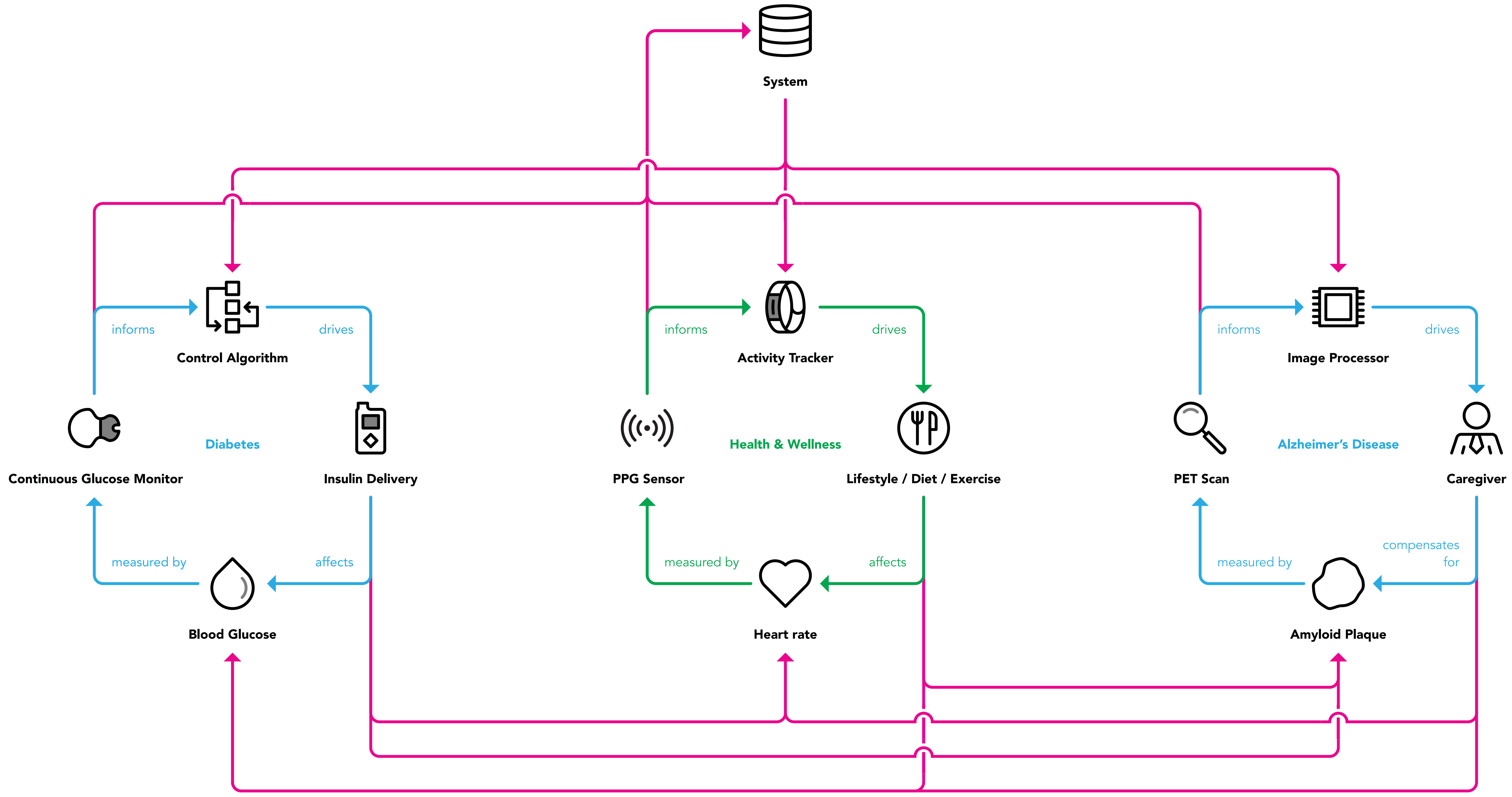
Hand-held Controller

Insulin Pump

Infusion Set

Patient

Integration of individual disease management systems in a larger ecology.



Process maps

A full explanation usually requires two frames:

- What a system ***is*** — a descriptive frame
  a focus on the elements, the nouns (a concept map)

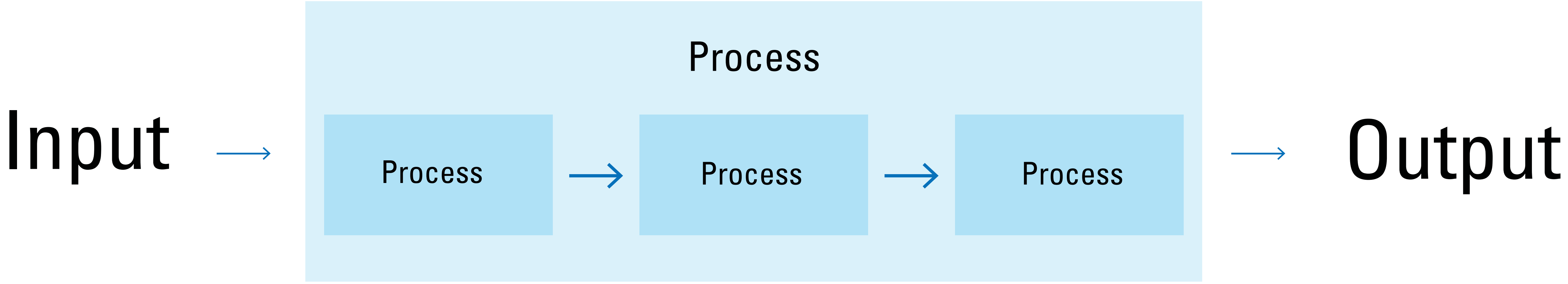
  and

- What the system ***does*** — a prescriptive frame
  a focus on actions, the verbs (a process map)

A process may be represented by a node with inputs + outputs.

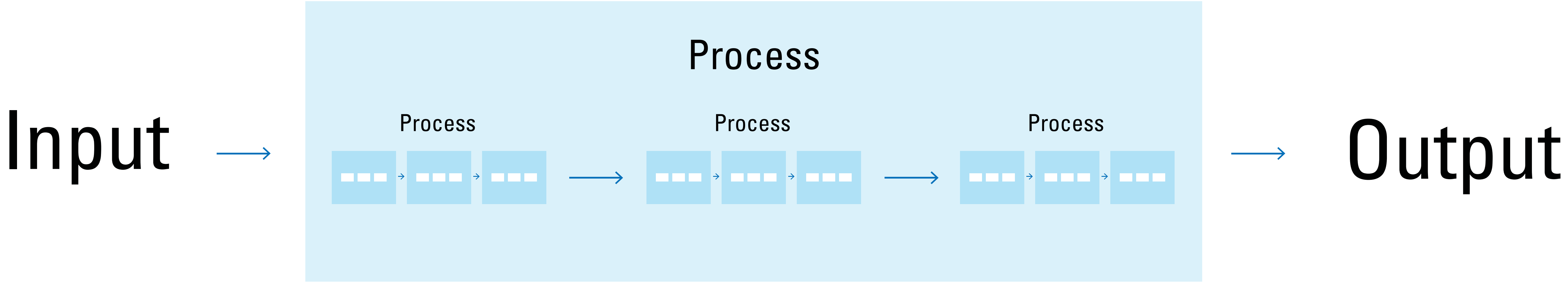
Input → Process → Output

The number of sub-processes shown is subjective; it depends on the context of use of the model.

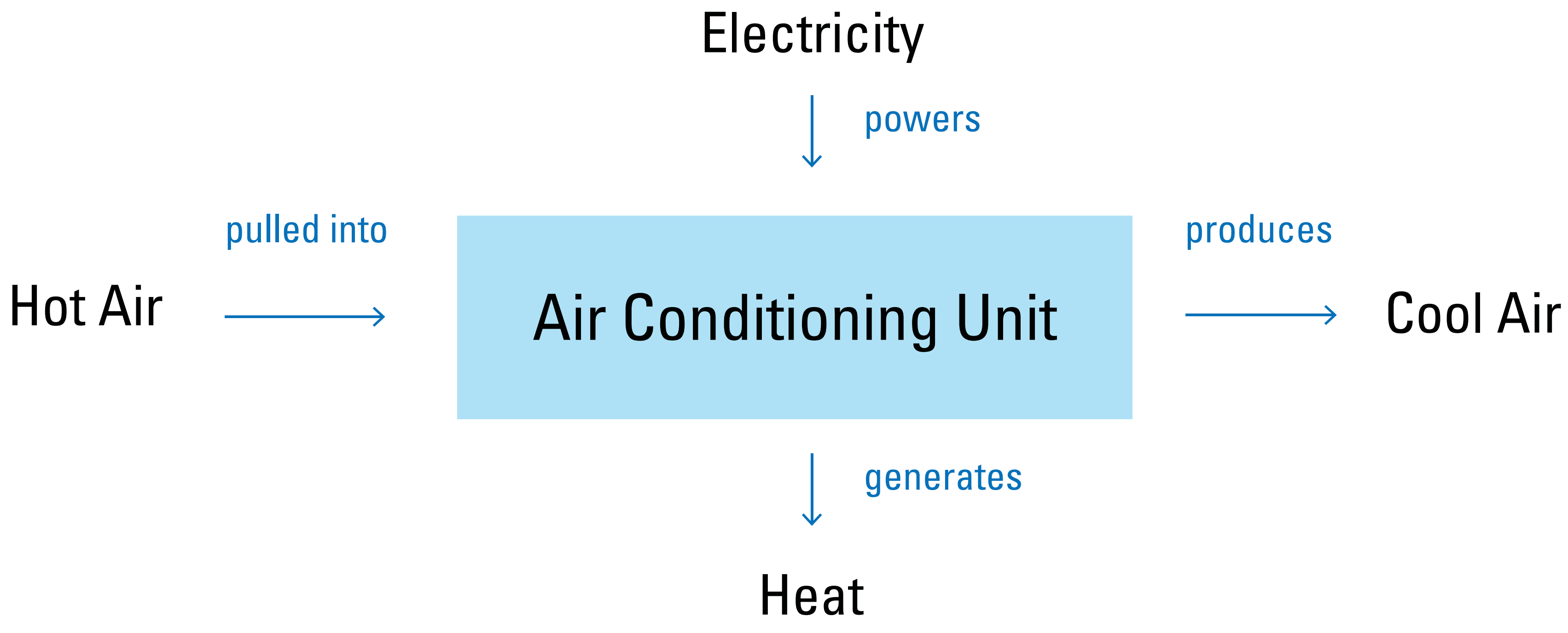


Input → Process (Process → Process → Process) → Output

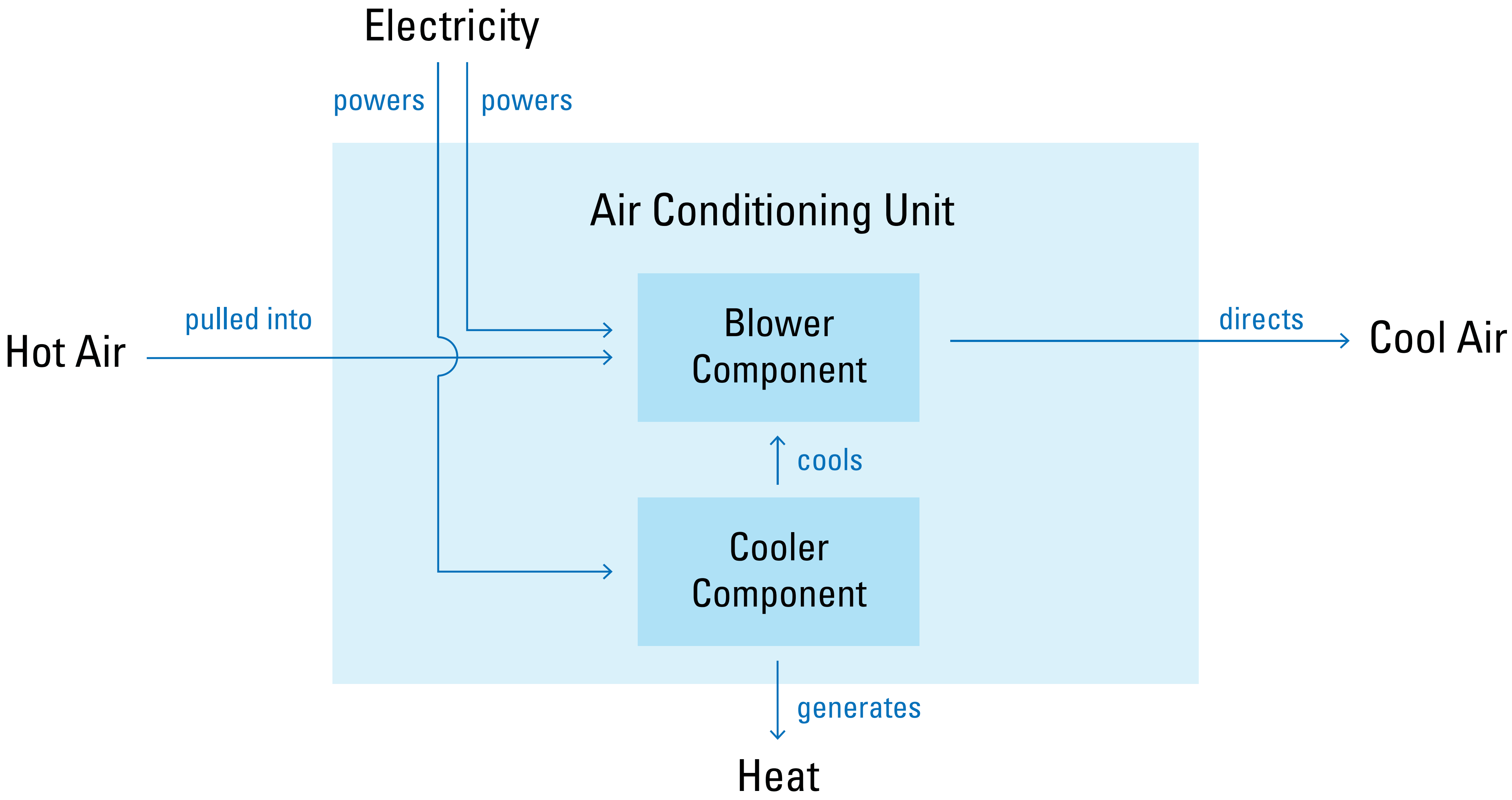
A process may be subdivided almost endlessly — not unlike Zeno's paradox.



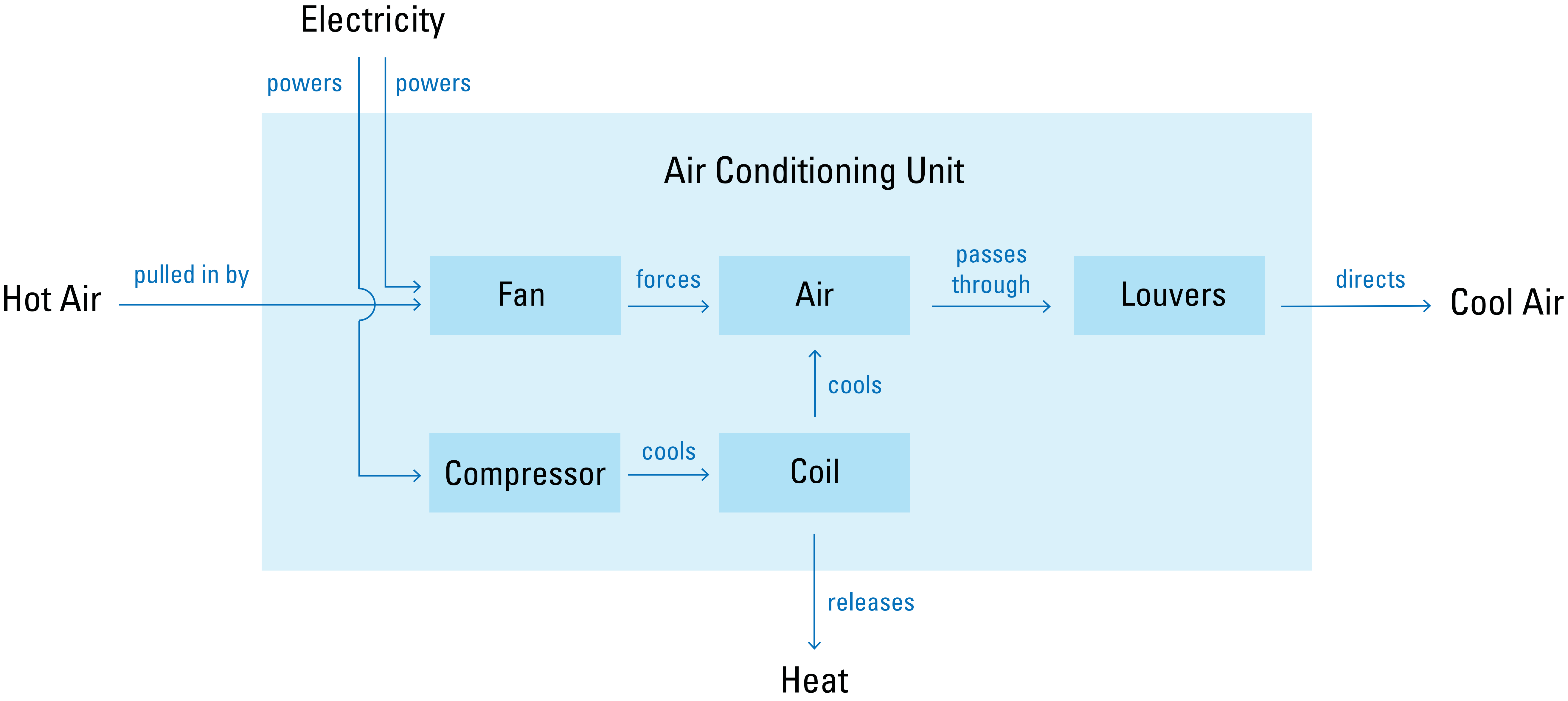
For example, an air conditioner takes in hot air + electricity and returns both cool air (inside) and heat (outside).



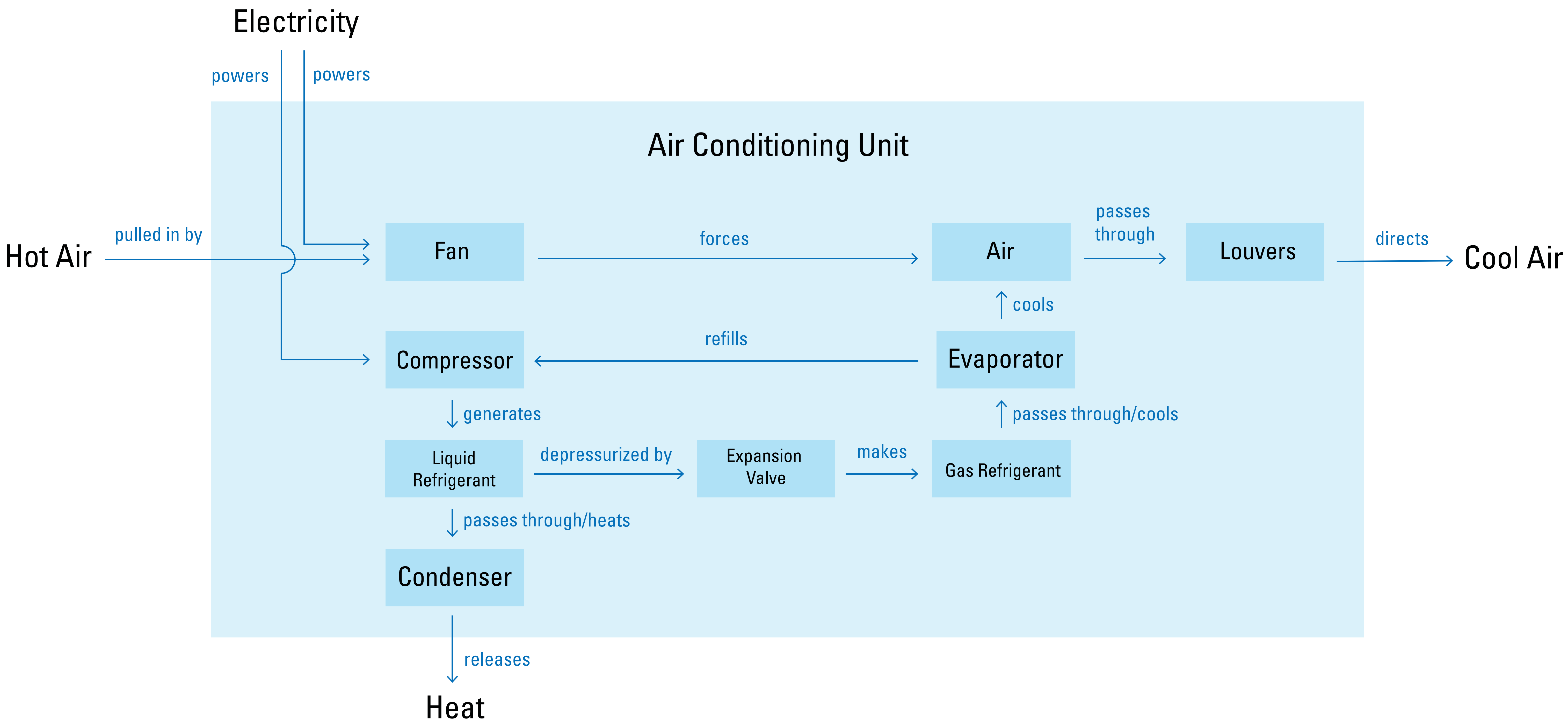
An air conditioner has two primary components: a blower unit and a cooler unit.



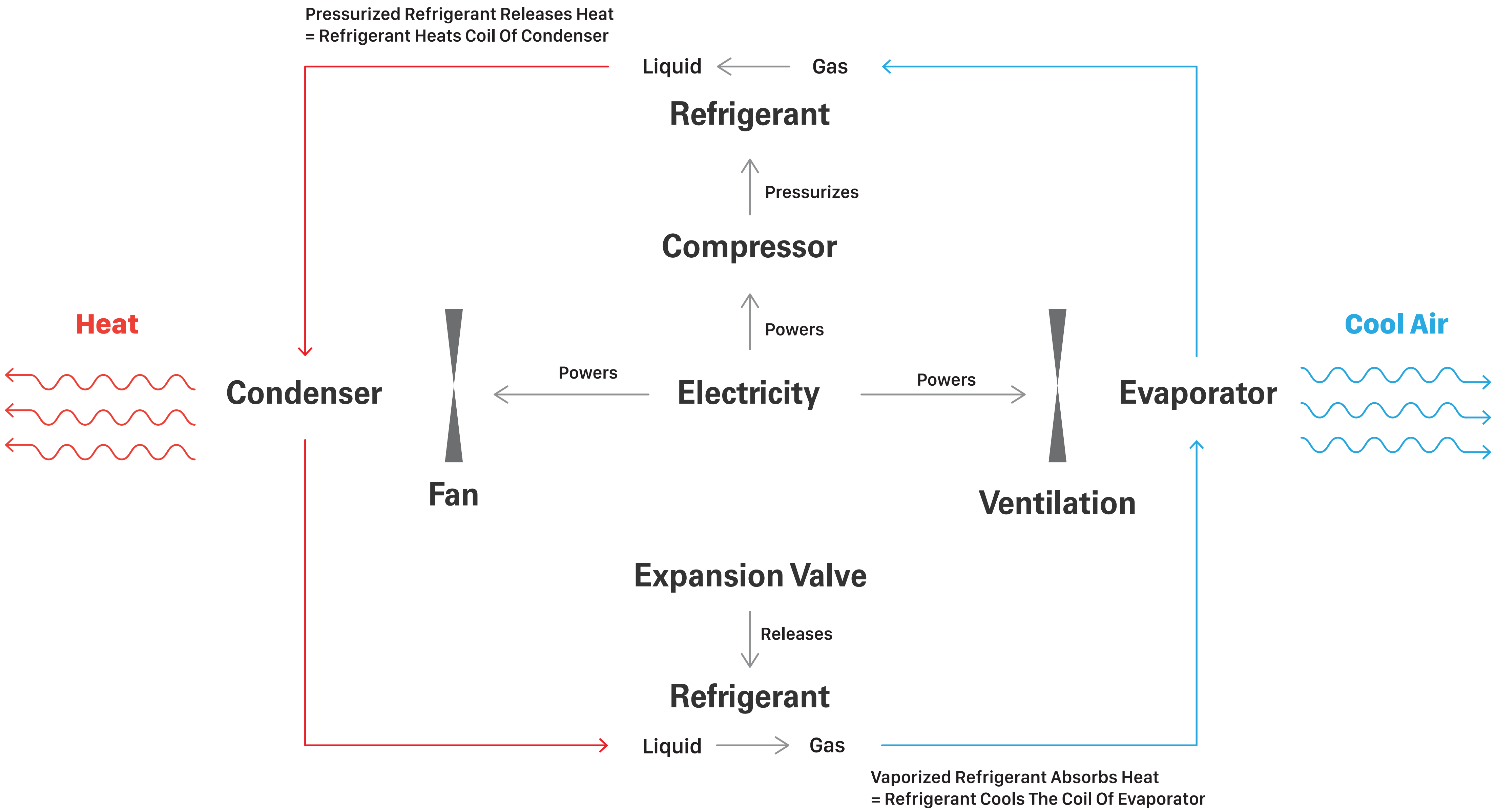
Zooming in, we see the blower includes a fan + louvers, and the cooler includes a compressor + cooling coil.



Zooming in further, we see the details of the coolant recycling loop.



Alternative representations may emphasize different parts of the process; here, the model emphasizes the coolant recycling loop.



Pressurized Refrigerant Releases Heat
= Refrigerant Heats Coil Of Condenser

Liquid ← Gas

Refrigerant

↑ Pressurizes

Compressor

↑ Powers

Heat

Condenser ← Powers — Electricity — Powers → Evaporator

Cool Air

Fan

Ventilation

Expansion Valve

↓ Releases

Refrigerant

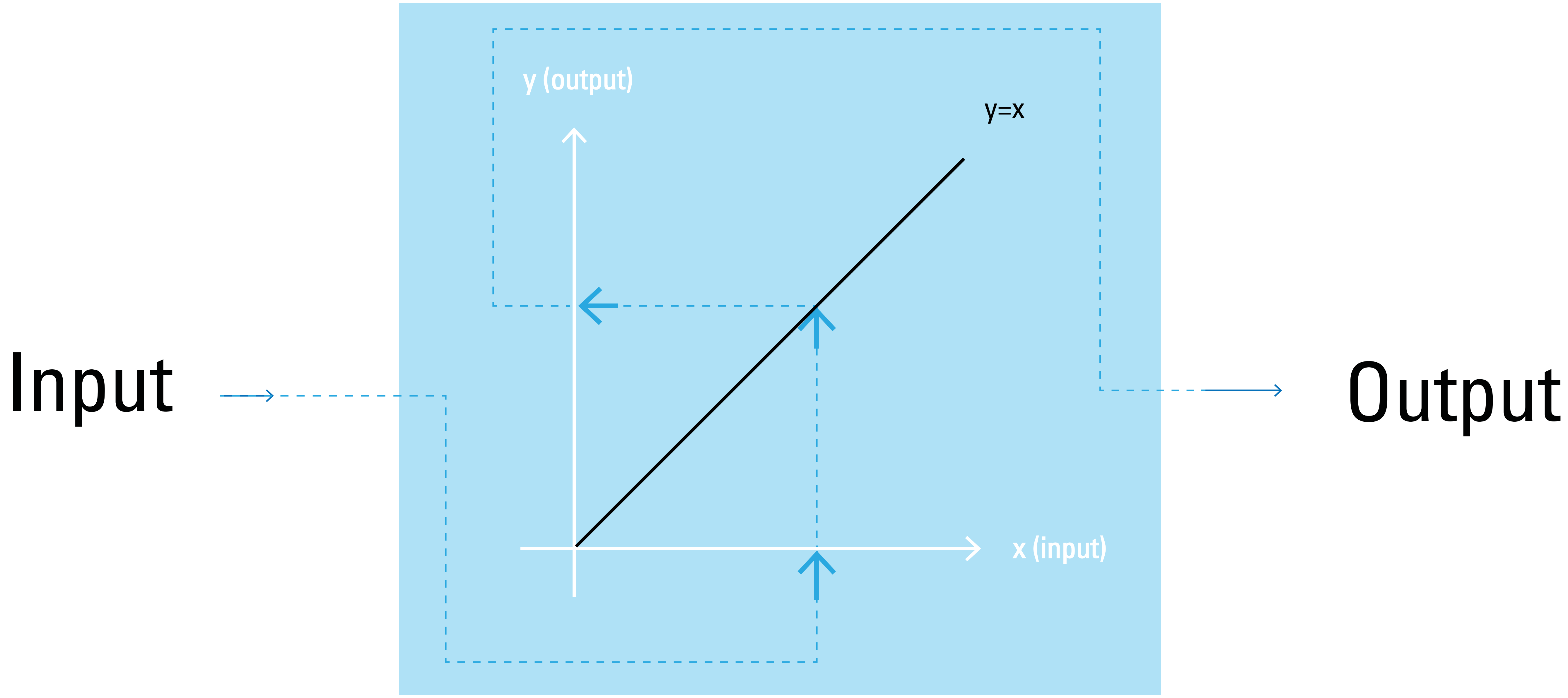
Liquid → Gas

Vaporized Refrigerant Absorbs Heat
= Refrigerant Cools The Coil Of Evaporator

A process taking an input and returning an output may also be thought of as a transform function.

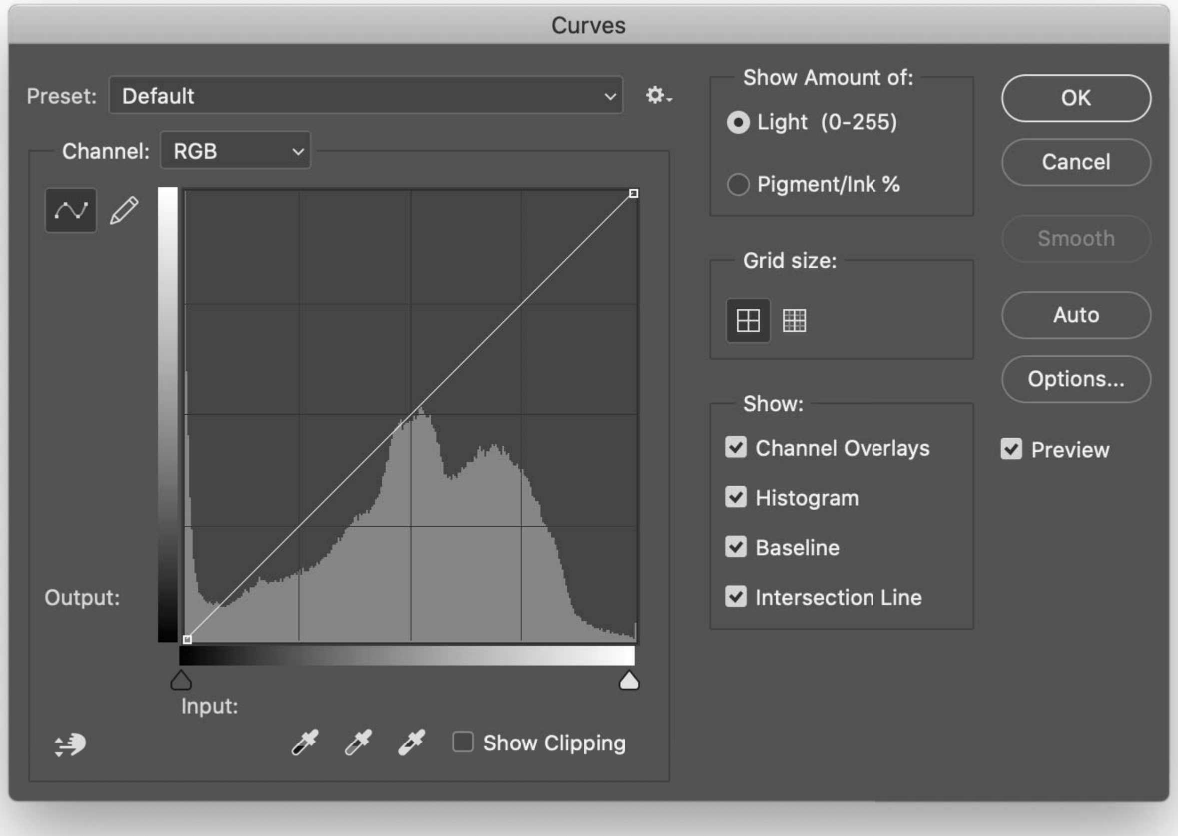
Input → Transform Function → Output

A transform function may be represented as a graph, i.e., $y = mx + b$.



Photoshop makes explicit use of this function.

Input →



Curves

Preset: Default

Channel: RGB

Show Amount of:
● Light (0-255)
○ Pigment/Ink %

Show:
☑ Channel Overlays
☑ Histogram
☑ Baseline
☑ Intersection Line

☑ Preview

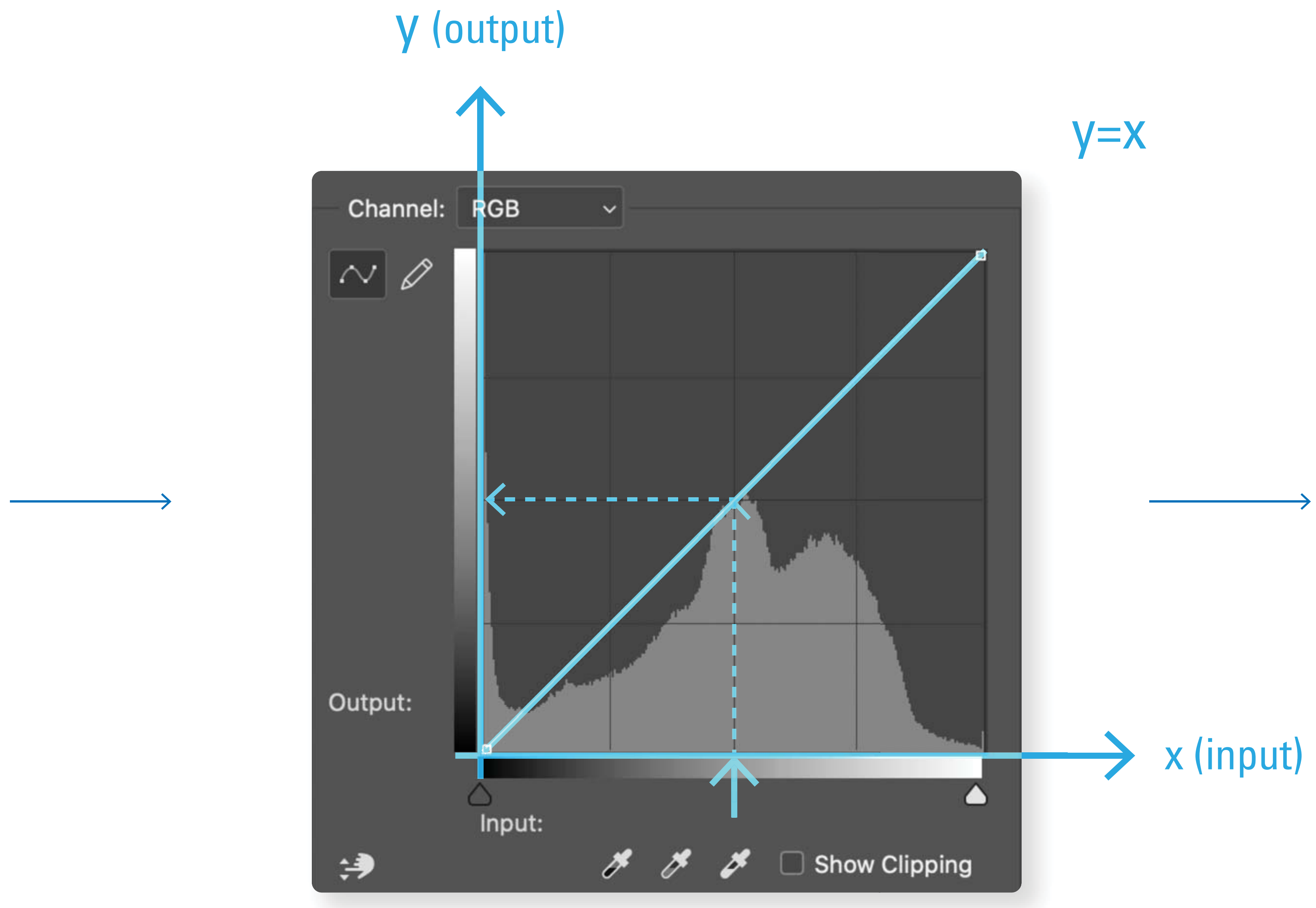
☐ Show Clipping

→ Output

Users may change the shape of the curve, thus changing the transform.

Input

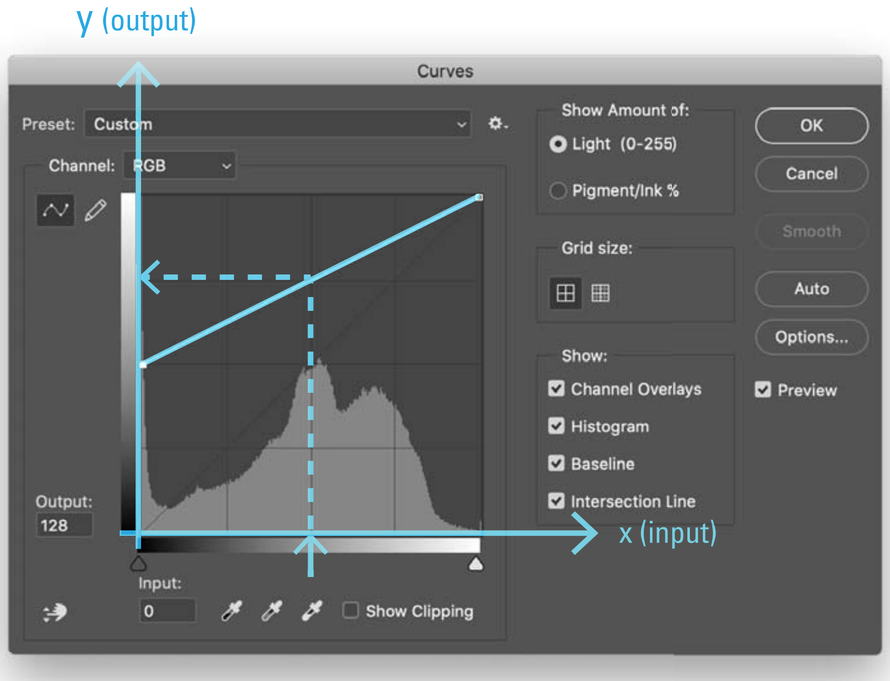
Output



Slope (m) determines contrast; y intercept (b) affects darkness.

Input

→



Preset: Custom

Channel: RGB

Show Amount of:
● Light (0-255)
○ Pigment/Ink %

☑ Channel Overlays
☑ Histogram
☑ Baseline
☑ Intersection Line

☑ Preview

Output: 128

Input: 0

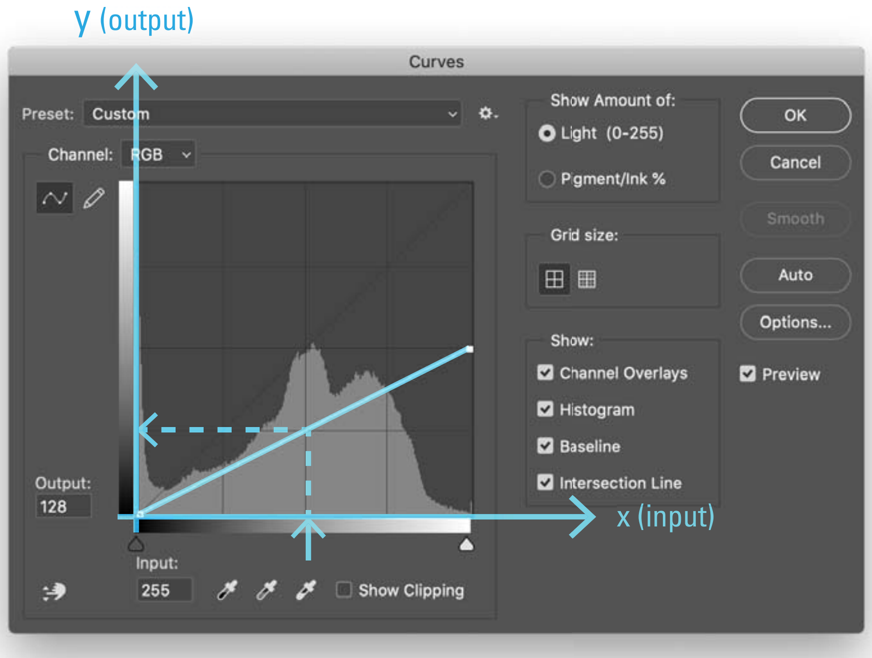
☐ Show Clipping

→

Output

Input

→



Preset: Custom

Channel: RGB

Show Amount of:
● Light (0-255)
○ Pigment/Ink %

☑ Channel Overlays
☑ Histogram
☑ Baseline
☑ Intersection Line

☑ Preview

Output: 128

Input: 255

☐ Show Clipping

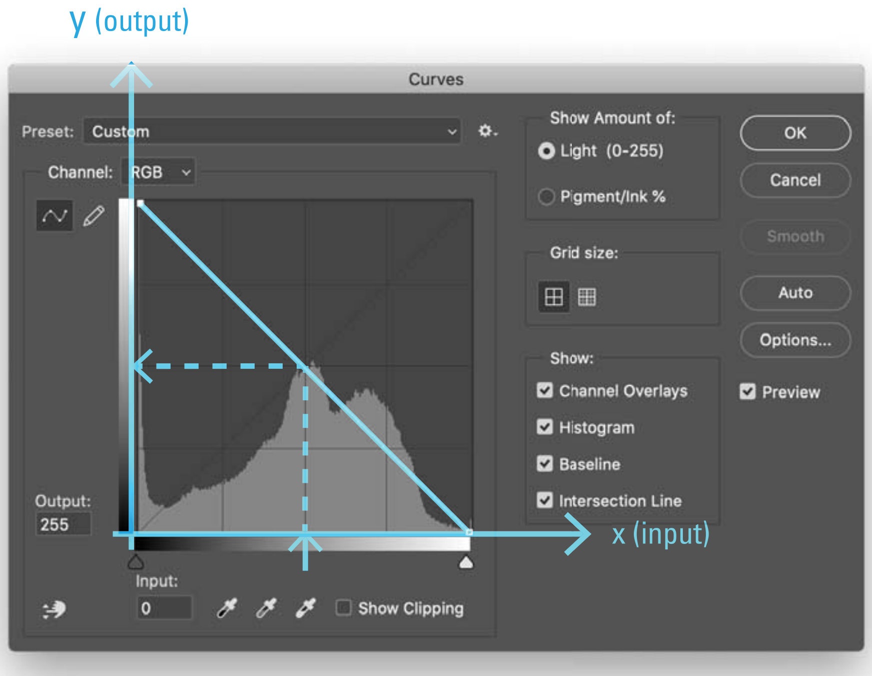
→

Output

y = -x returns the negative image.

Input

→



Preset: Custom

Channel: RGB

Show Amount of:
● Light (0-255)
○ Pigment/Ink %

Grid size:

Show:
☑ Channel Overlays
☑ Histogram
☑ Baseline
☑ Intersection Line

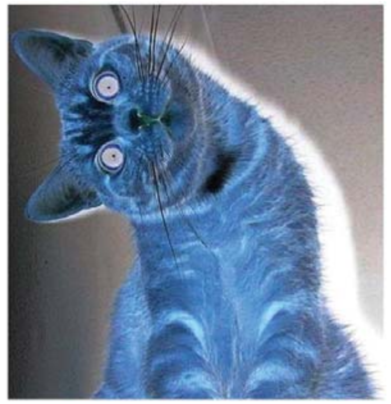
Output: 255

Input: 0

☐ Show Clipping

☑ Preview

→

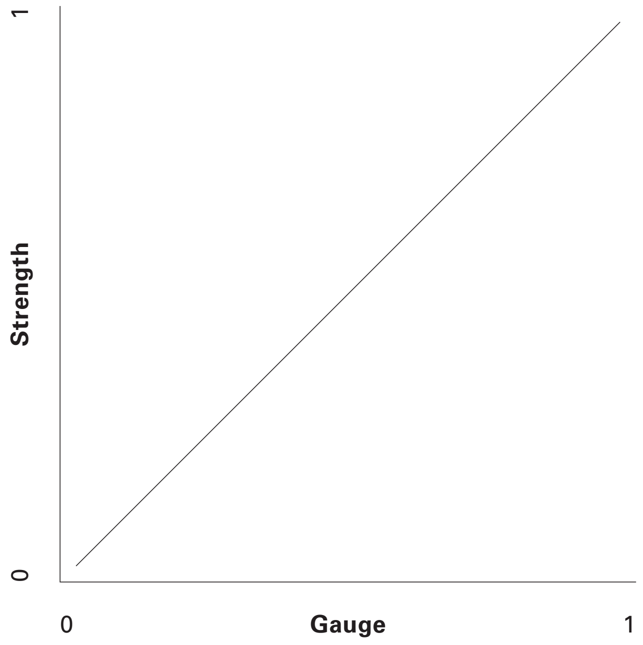


Output

We used these ideas in the conceptual design of a UI for a tool for folks who design auto-injectors.

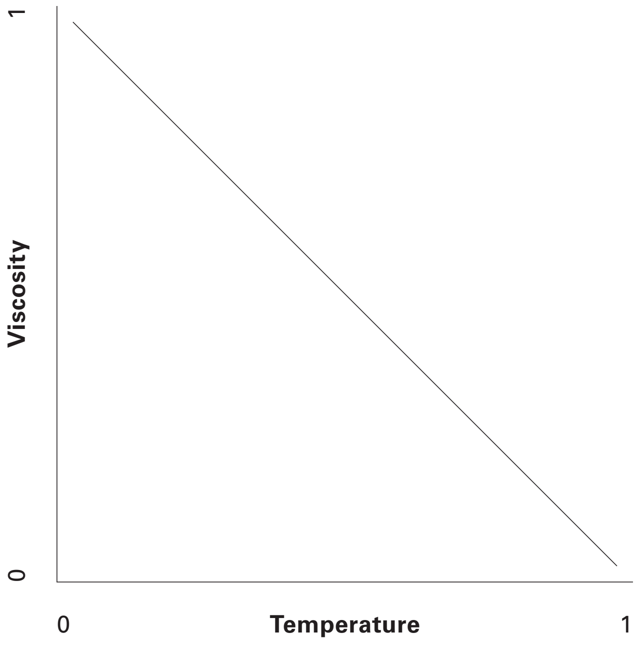
Proportional Relationships

For example, needle strength is proportional to Gauge. Thin needles are weaker than thick ones.



Inverse Relationships

For example, the viscosity of a compound is inversely proportional to temperature. Cold compounds are thicker and more viscous then warm ones.

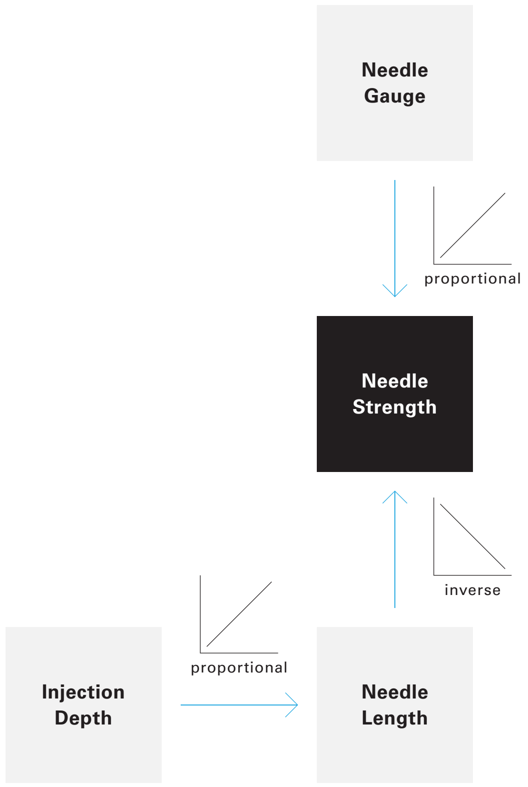


Deeper injections require higher gauge needles, because they have to be longer and thus stronger.

Length is proportional to depth.
Strength is inverse to length.
Strength is proportional to gauge.

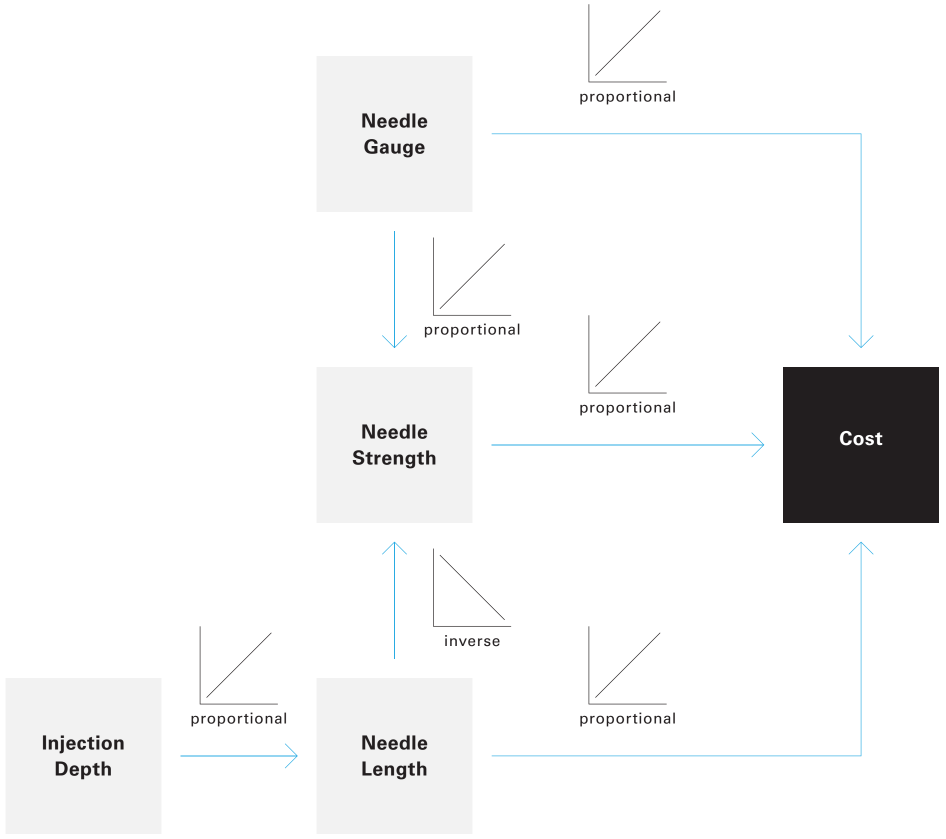
As depth increases, length increases.
As length increases, strength decreases,
meaning risk of breaking increases.

Thus strength must be maintained—
typically by increasing gauge.

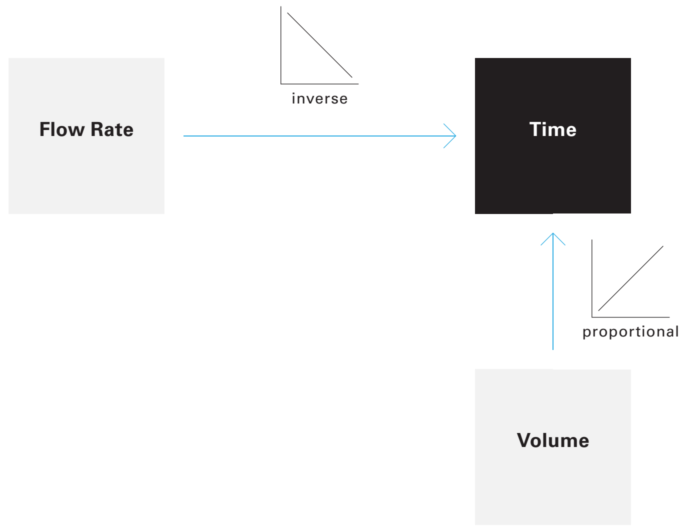


Strength and gauge affect cost.

Gauge, strength, and length also affect cost.
Changing materials may affect relationships.



Time (length of injection) is also a factor.

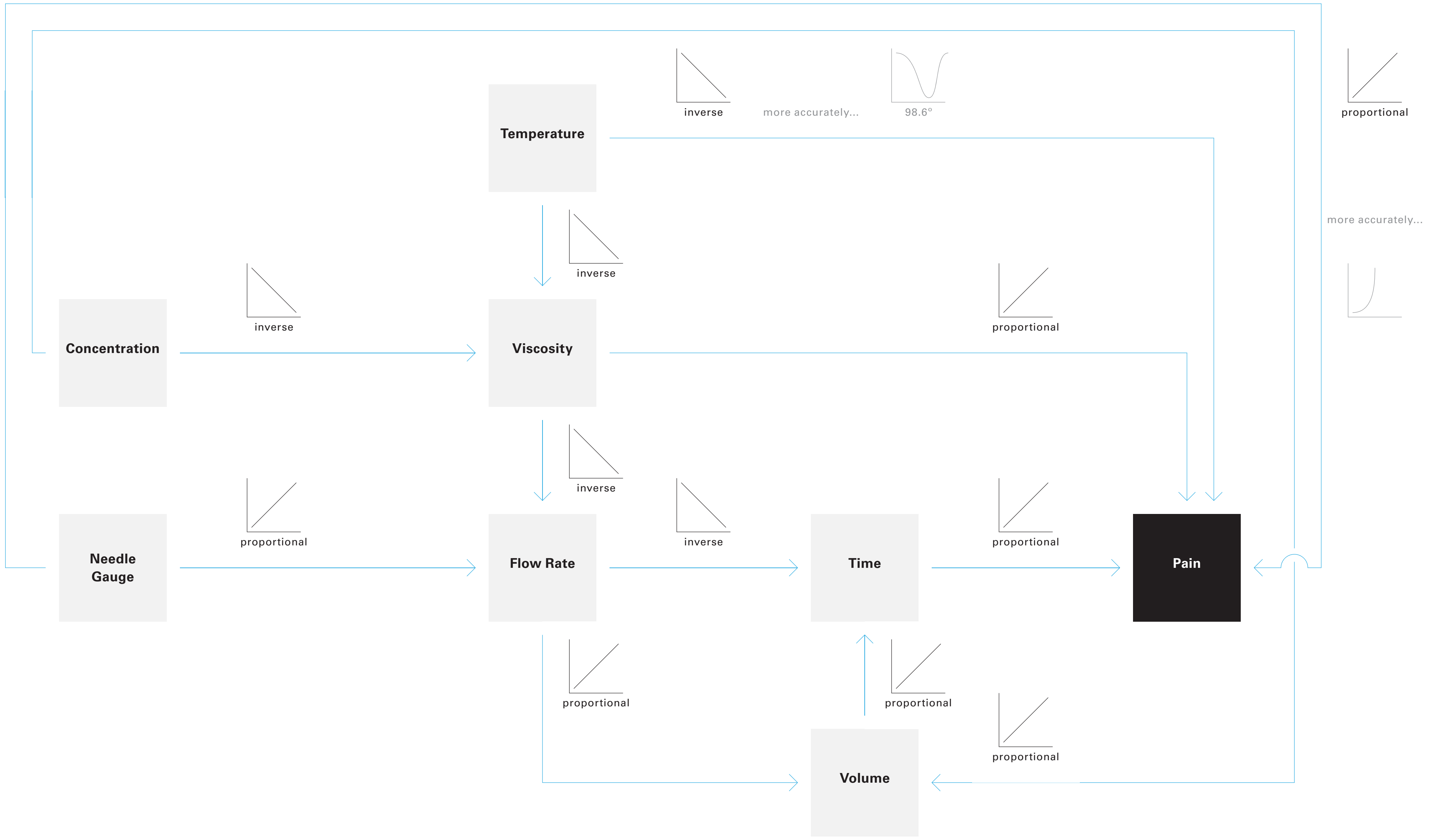


Time = $\frac{\text{Volume}}{\text{Flow Rate}}$

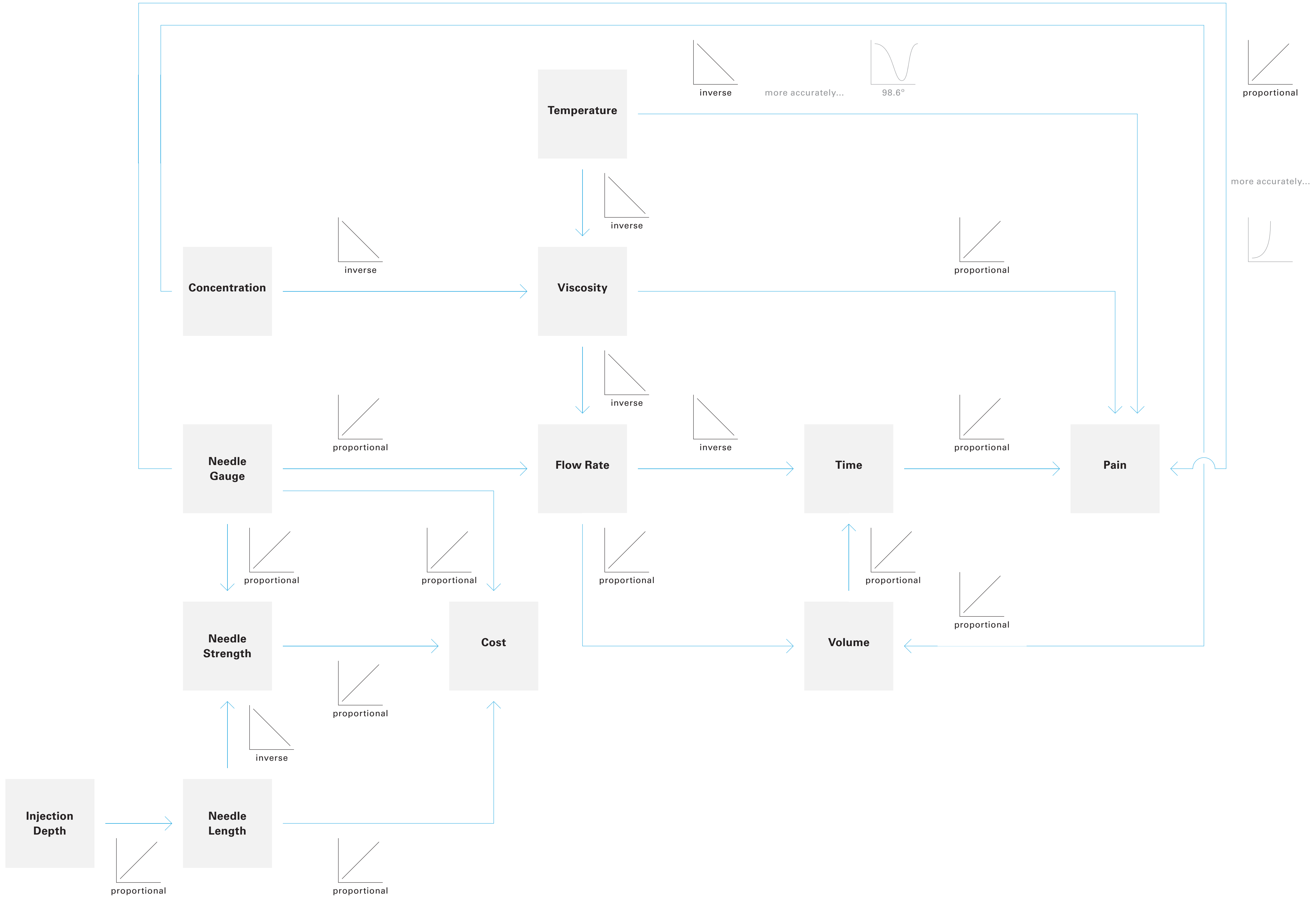
Volume = Time × Flow Rate

Flow Rate = $\frac{\text{Volume}}{\text{Time}}$

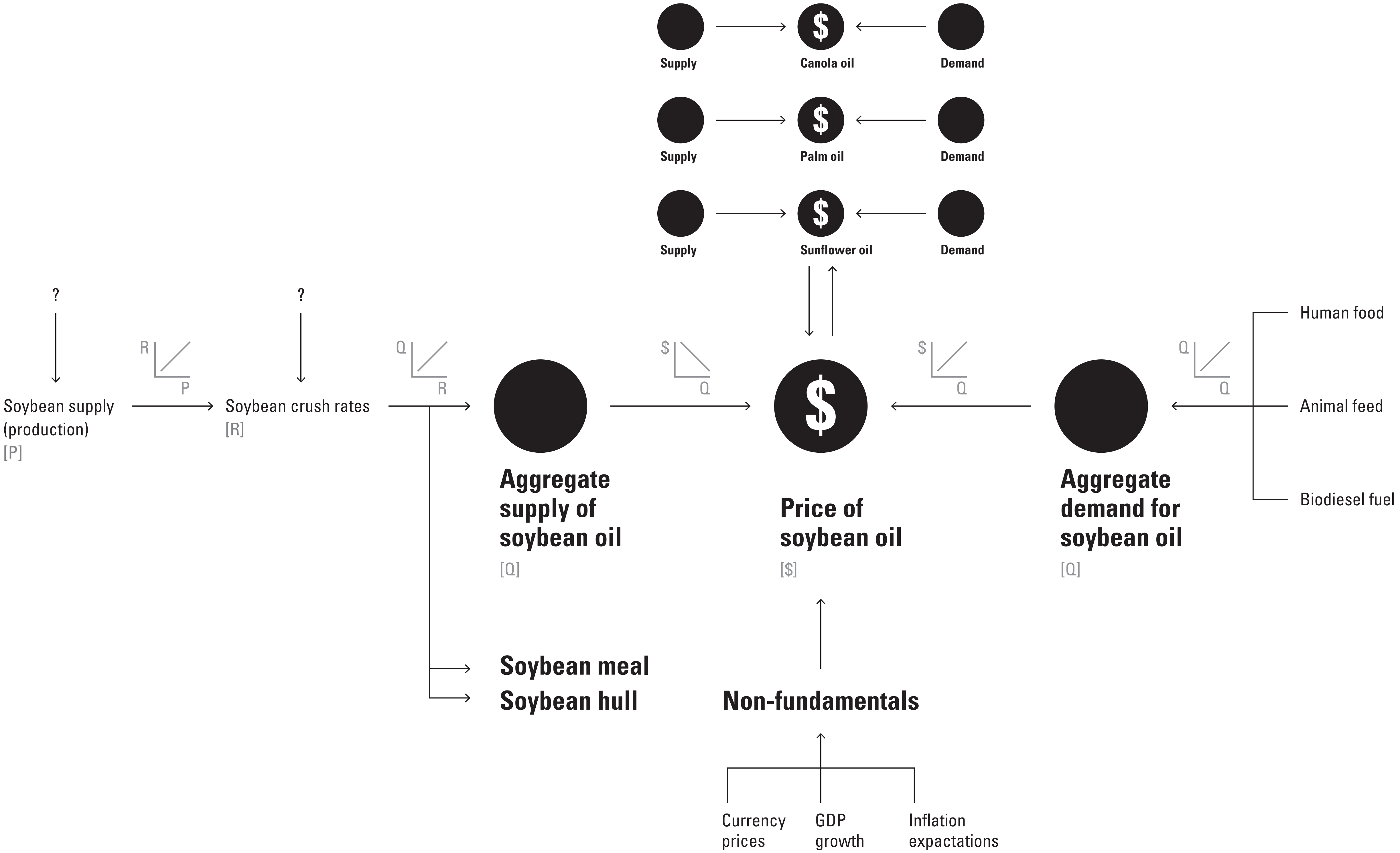
Patient pain is affected by time, viscosity, and temperature.



The factors affecting AI needle design and their relationships.



The transform function is integral to understanding supply and demand.



Supply
Canola oil
Demand

Supply
Palm oil
Demand

Supply
Sunflower oil
Demand

?
?

Soybean supply
(production)
[P]

Soybean crush rates
[R]

Aggregate supply of soybean oil
[Q]

Price of soybean oil
[$]

Aggregate demand for soybean oil
[Q]

Human food
Animal feed
Biodiesel fuel

Soybean meal
Soybean hull

Non-fundamentals

Currency prices
GDP growth
Inflation expactations

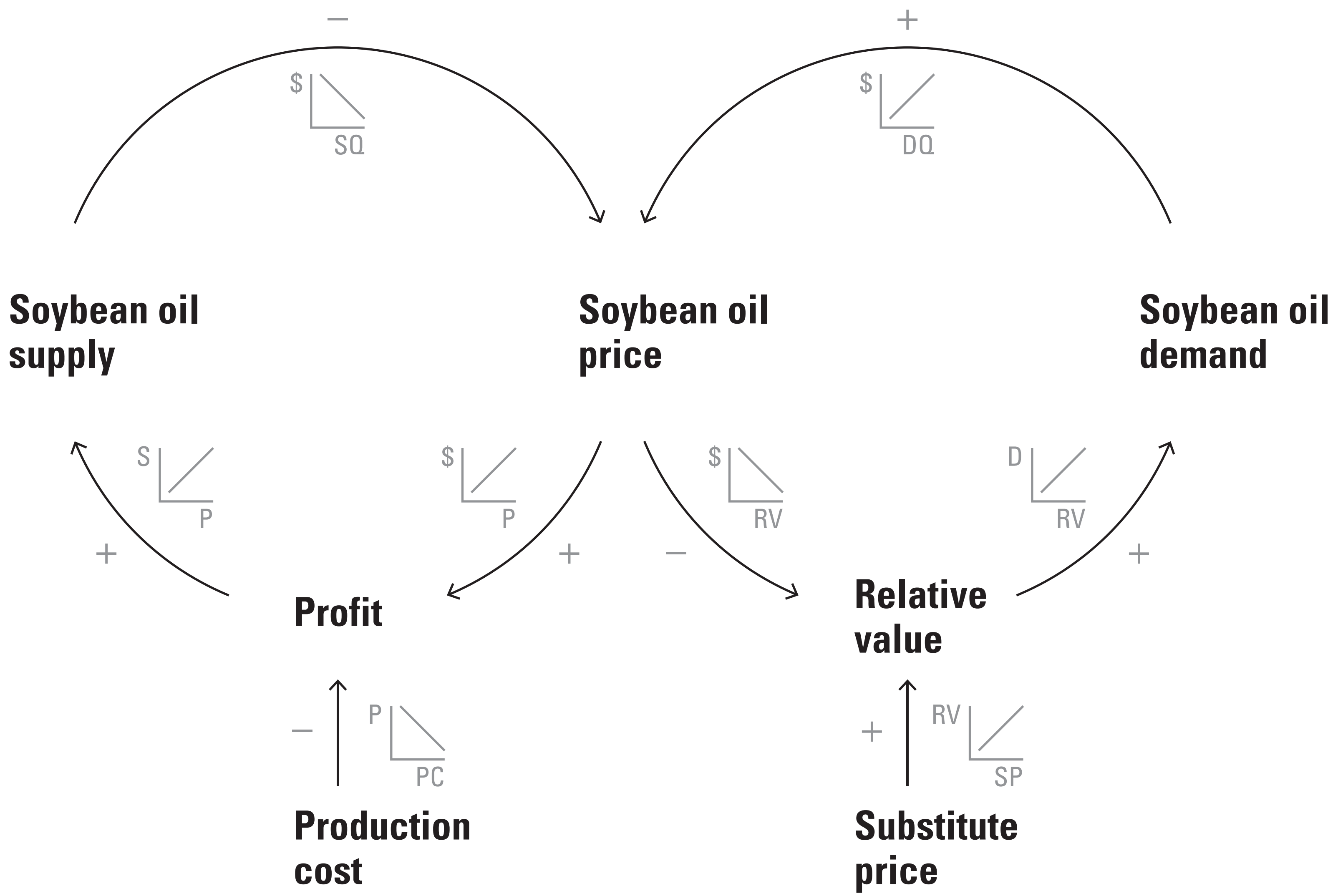
An alternative visualization of supply and demand.

If supply decreases,
then price increases.

If price increases,
then profit increases.

If profit increases,
then supply increases.

If production cost decreases,
then profit increases.



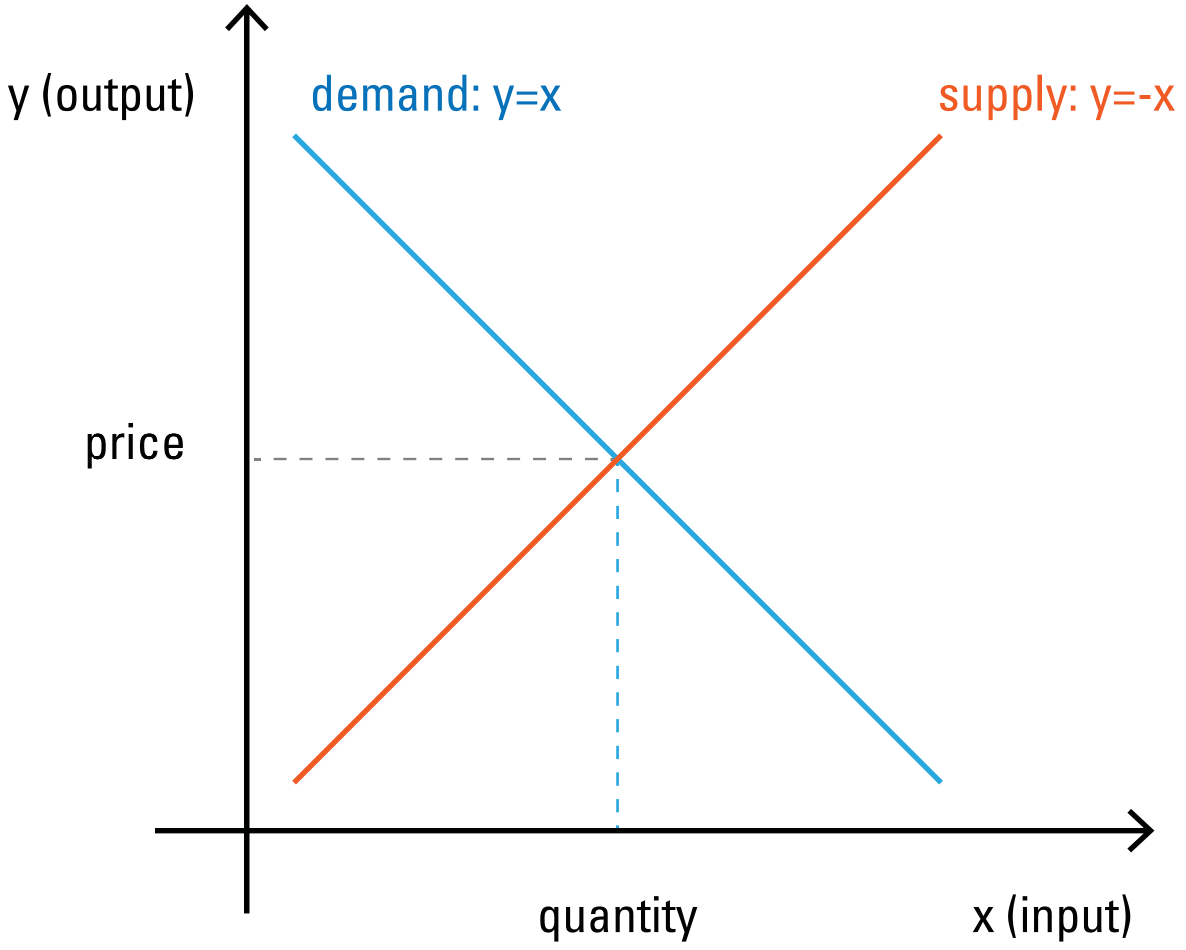
If demand increases,
then price increases.

If price decreases,
then relative value increases.

If relative value increases,
then demand increases.

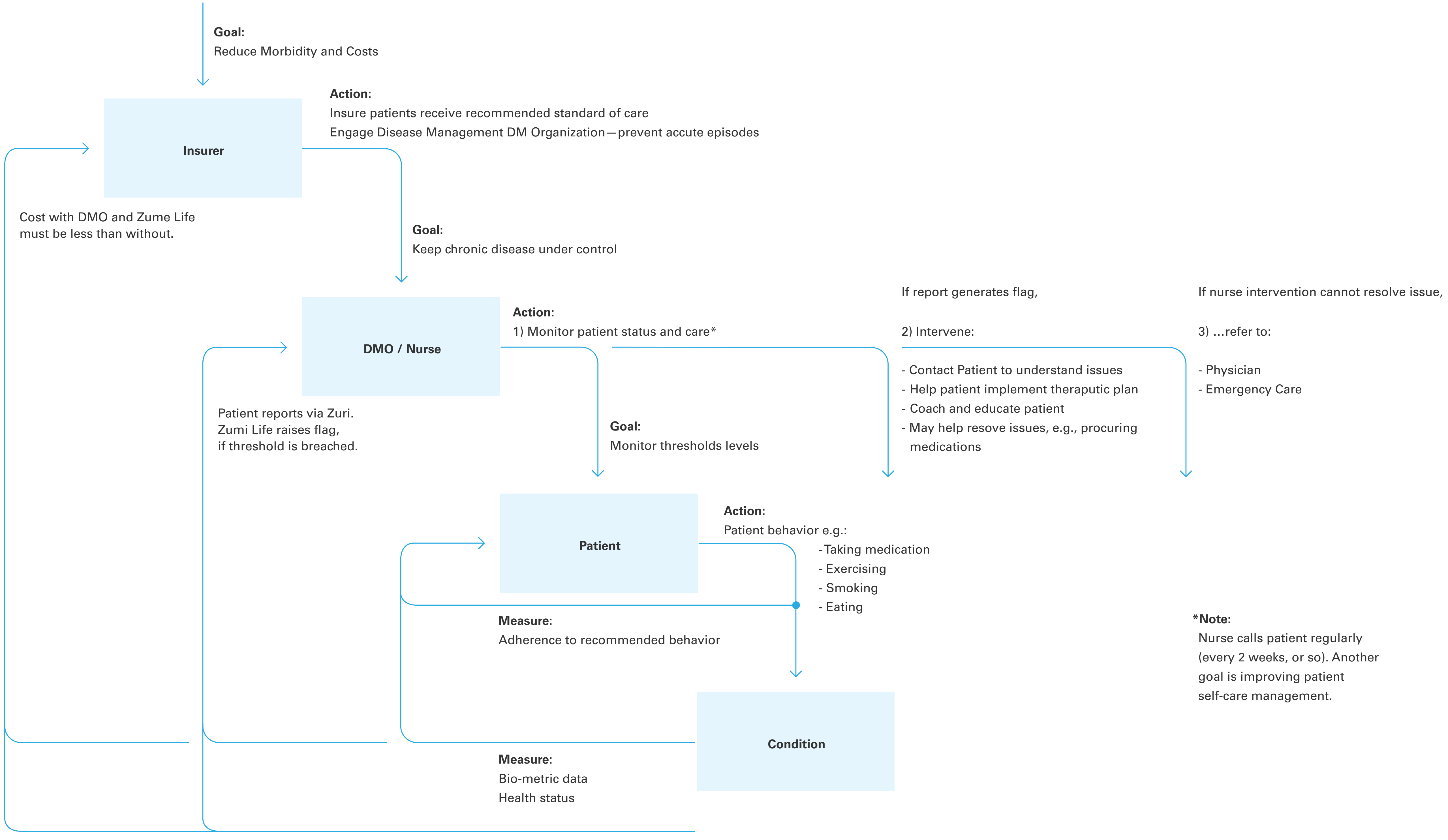
If substitute price increases,
then relative value increases.

Price equilibrium is defined as the point where supply = demand.



Case study:
Health self-management service

Health self-management service: Goal structure



Goal:
Reduce Morbidity and Costs

Insurer

Action:
Insure patients receive recommended standard of care
Engage Disease Management DM Organization—prevent accute episodes

Cost with DMO and Zume Life
must be less than without.

Goal:
Keep chronic disease under control

DMO / Nurse

Action:
1) Monitor patient status and care*

If report generates flag,

2) Intervene:

- Contact Patient to understand issues
- Help patient implement theraputic plan
- Coach and educate patient
- May help resove issues, e.g., procuring medications

If nurse intervention cannot resolve issue,

3) …refer to:

- Physician
- Emergency Care

Patient reports via Zuri.
Zumi Life raises flag,
if threshold is breached.

Goal:
Monitor thresholds levels

Patient

Action:
Patient behavior e.g.:
- Taking medication
- Exercising
- Smoking
- Eating

Measure:
Adherence to recommended behavior

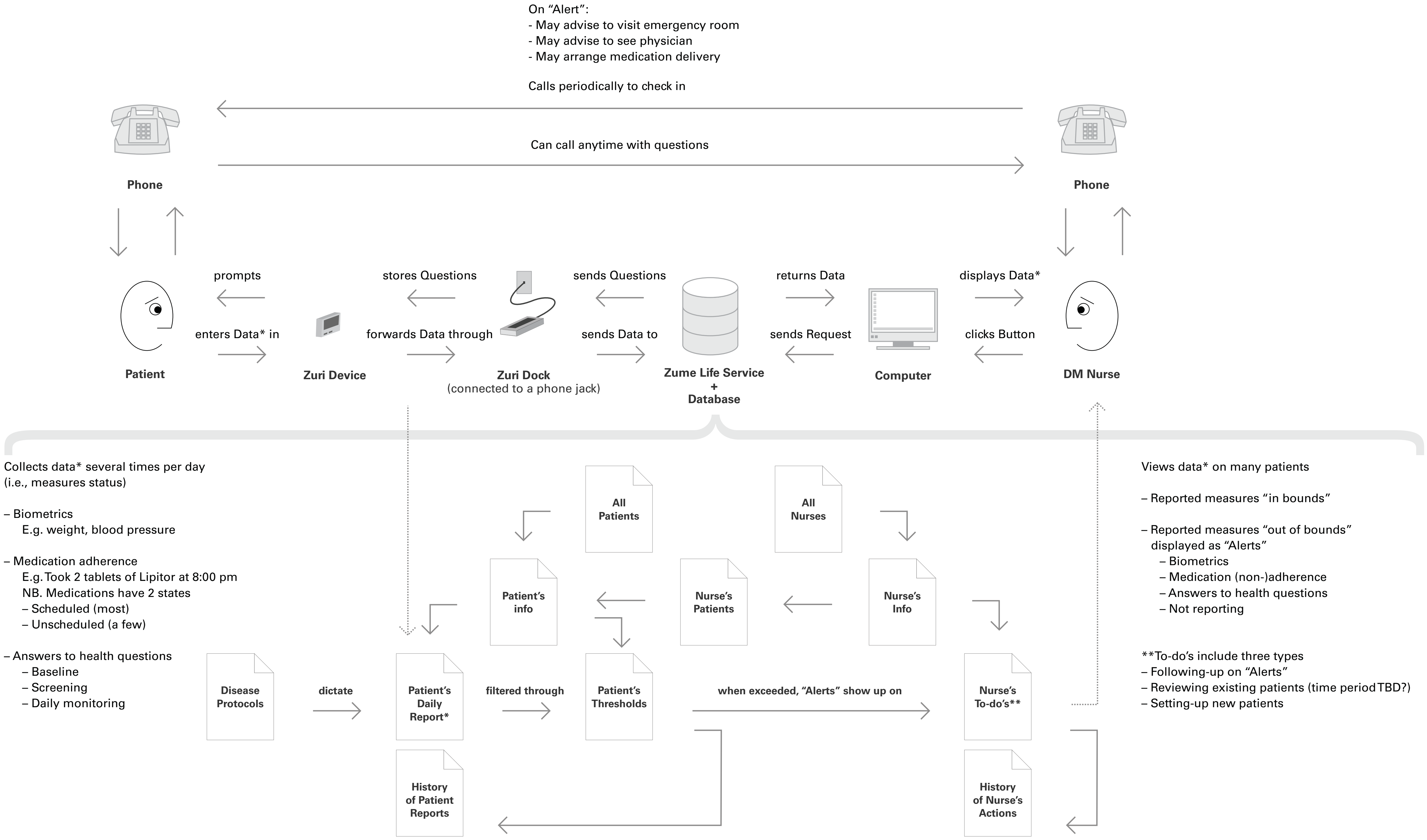
Condition

Measure:
Bio-metric data
Health status

Measure:
Cost of treatments

***Note:**
Nurse calls patient regularly (every 2 weeks, or so). Another goal is improving patient self-care management.

Health self-management service: Data structure



On "Alert":
- May advise to visit emergency room
- May advise to see physician
- May arrange medication delivery

Calls periodically to check in

Can call anytime with questions

Phone

Phone

prompts

enters Data* in

stores Questions

forwards Data through

sends Questions

sends Data to

returns Data

sends Request

displays Data*

clicks Button

Patient

Zuri Device

Zuri Dock
(connected to a phone jack)

Zume Life Service
+
Database

Computer

DM Nurse

Collects data* several times per day
(i.e., measures status)

- Biometrics
  E.g. weight, blood pressure

- Medication adherence
  E.g. Took 2 tablets of Lipitor at 8:00 pm
  NB. Medications have 2 states
  - Scheduled (most)
  - Unscheduled (a few)

- Answers to health questions
  - Baseline
  - Screening
  - Daily monitoring

All Patients

All Nurses

Patient's info

Nurse's Patients

Nurse's Info

Disease Protocols

dictate

Patient's Daily Report*

filtered through

Patient's Thresholds

when exceeded, "Alerts" show up on

Nurse's To-do's**

History of Patient Reports

History of Nurse's Actions

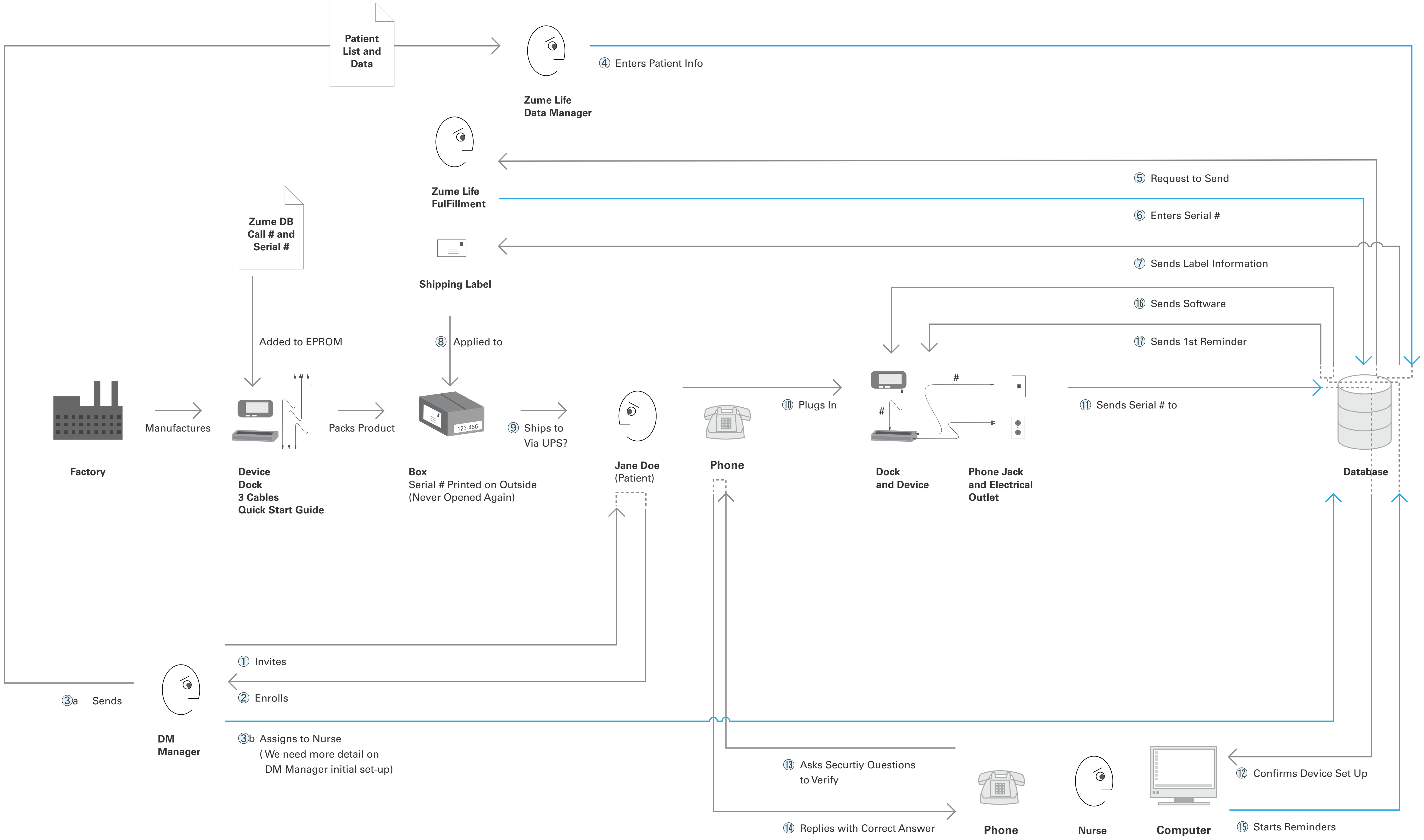
Views data* on many patients

- Reported measures "in bounds"

- Reported measures "out of bounds"
  displayed as "Alerts"
  - Biometrics
  - Medication (non-)adherence
  - Answers to health questions
  - Not reporting

**To-do's include three types
- Following-up on "Alerts"
- Reviewing existing patients (time period TBD?)
- Setting-up new patients

Health self-management service: Set-up process



Patient List and Data

④ Enters Patient Info

Zume Life Data Manager

Zume Life FulFillment

⑤ Request to Send

⑥ Enters Serial #

Zume DB Call # and Serial #

Shipping Label

⑦ Sends Label Information

⑯ Sends Software

⑰ Sends 1st Reminder

Added to EPROM

⑧ Applied to

Factory

Manufactures

Device
Dock
3 Cables
Quick Start Guide

Packs Product

Box
Serial # Printed on Outside
(Never Opened Again)

⑨ Ships to
Via UPS?

Jane Doe
(Patient)

Phone

⑩ Plugs In

Dock
and Device

Phone Jack
and Electrical
Outlet

⑪ Sends Serial # to

Database

① Invites

② Enrolls

③a Sends

DM
Manager

③b Assigns to Nurse
(We need more detail on
DM Manager initial set-up)

⑬ Asks Securtiy Questions
to Verify

⑭ Replies with Correct Answer

Phone

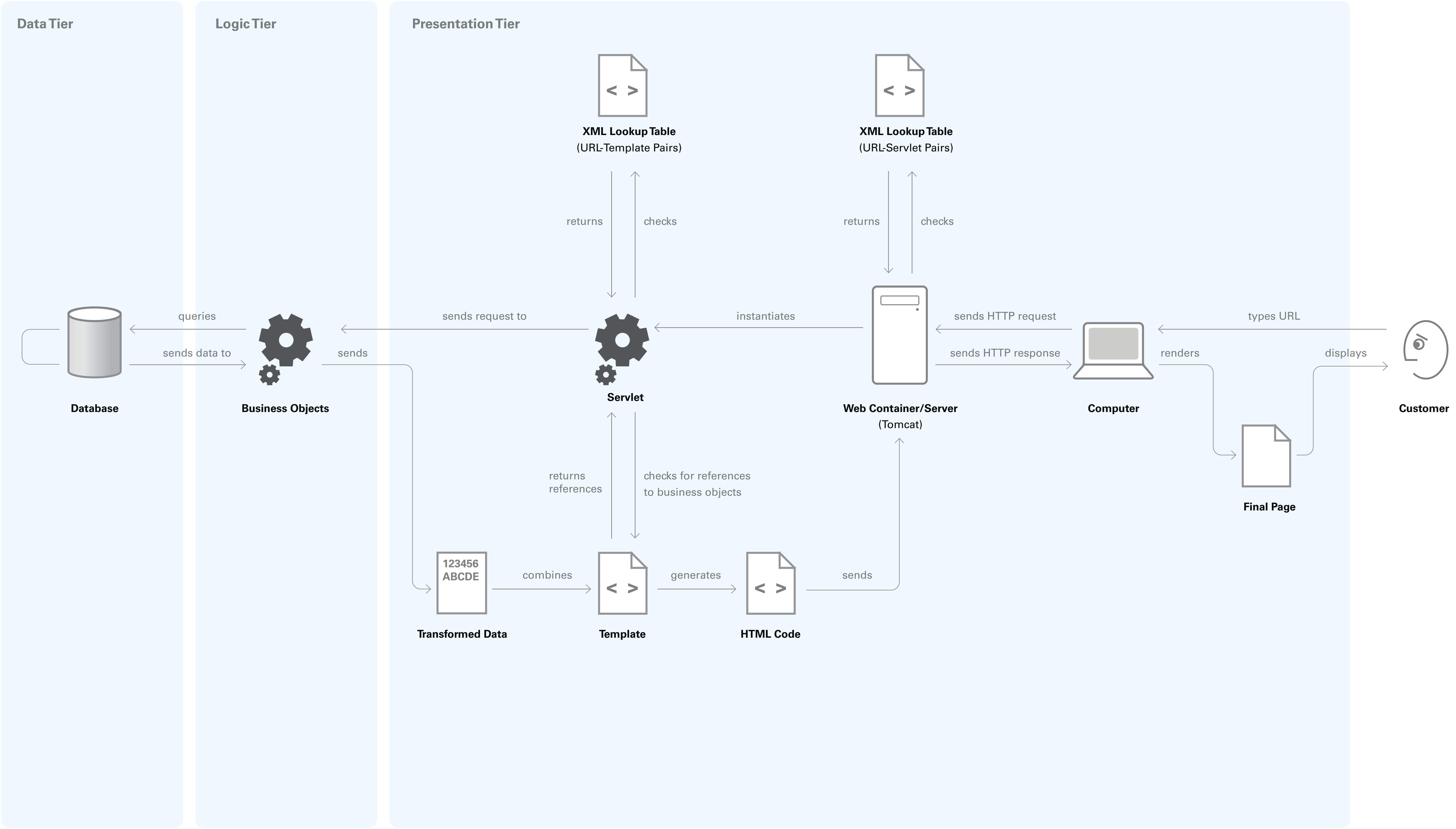
Nurse

Computer

⑫ Confirms Device Set Up

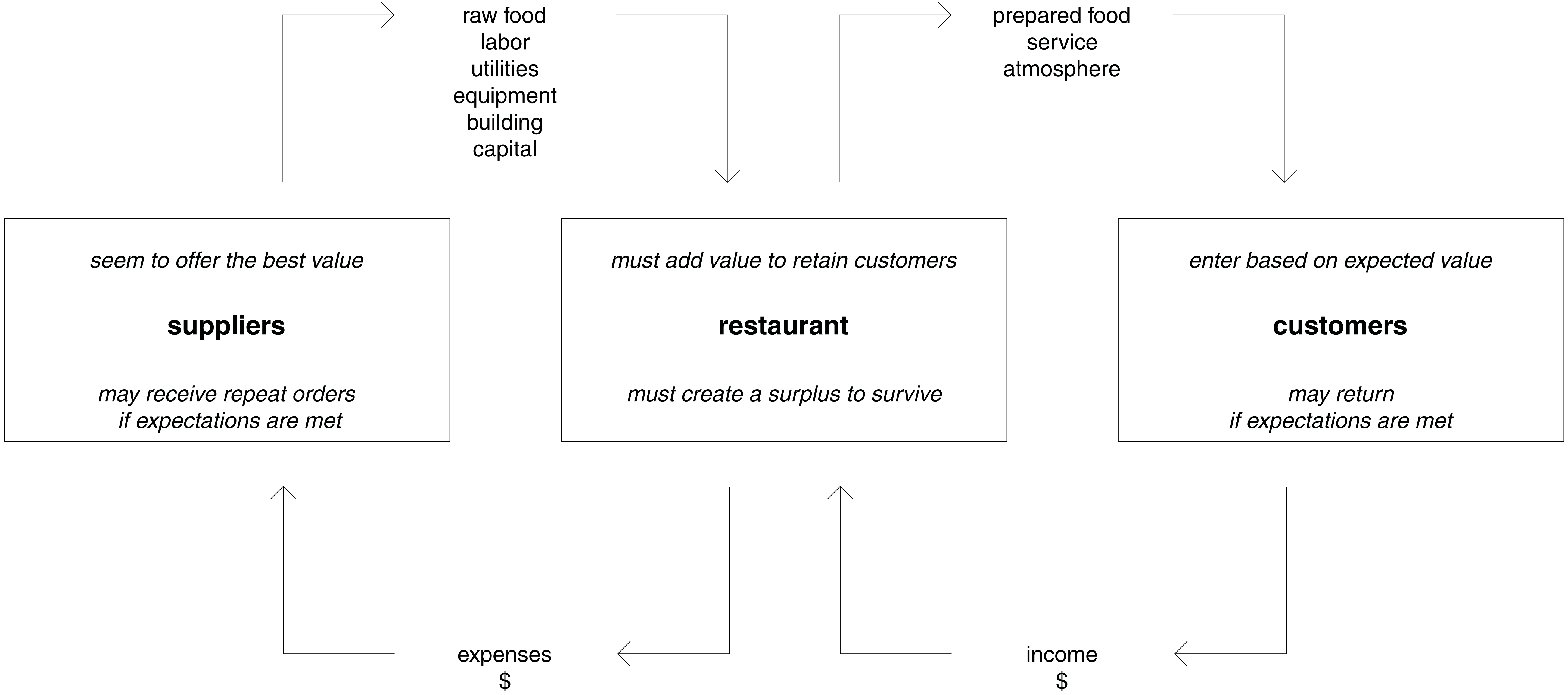
⑮ Starts Reminders

Health self-management service: Application architecture



Case study: Restaurants

Restaurant: Basic economic view



raw food
labor
utilities
equipment
building
capital

prepared food
service
atmosphere

seem to offer the best value

suppliers

may receive repeat orders
if expectations are met

must add value to retain customers

restaurant

must create a surplus to survive

enter based on expected value

customers

may return
if expectations are met

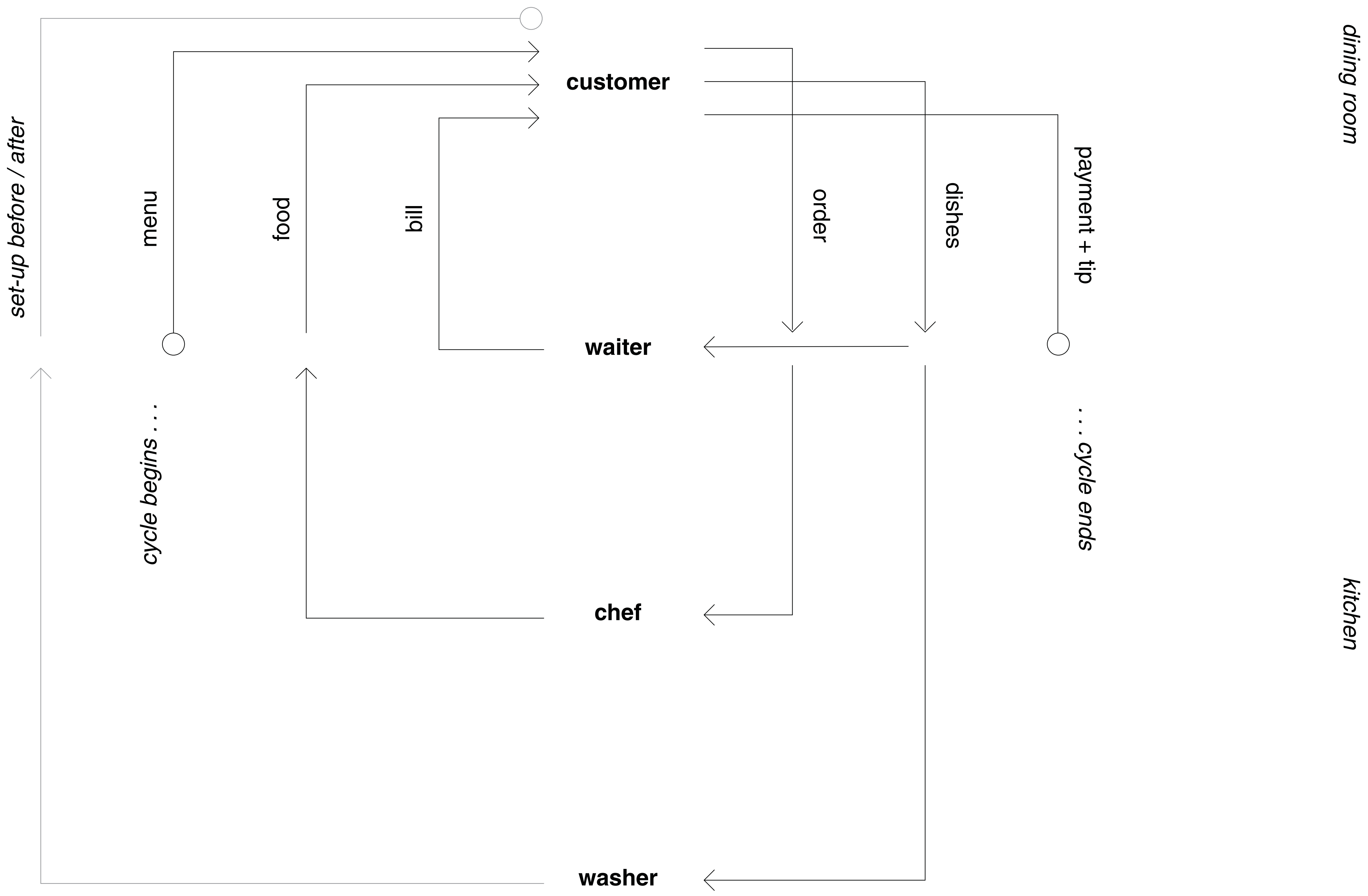
expenses
$

income
$

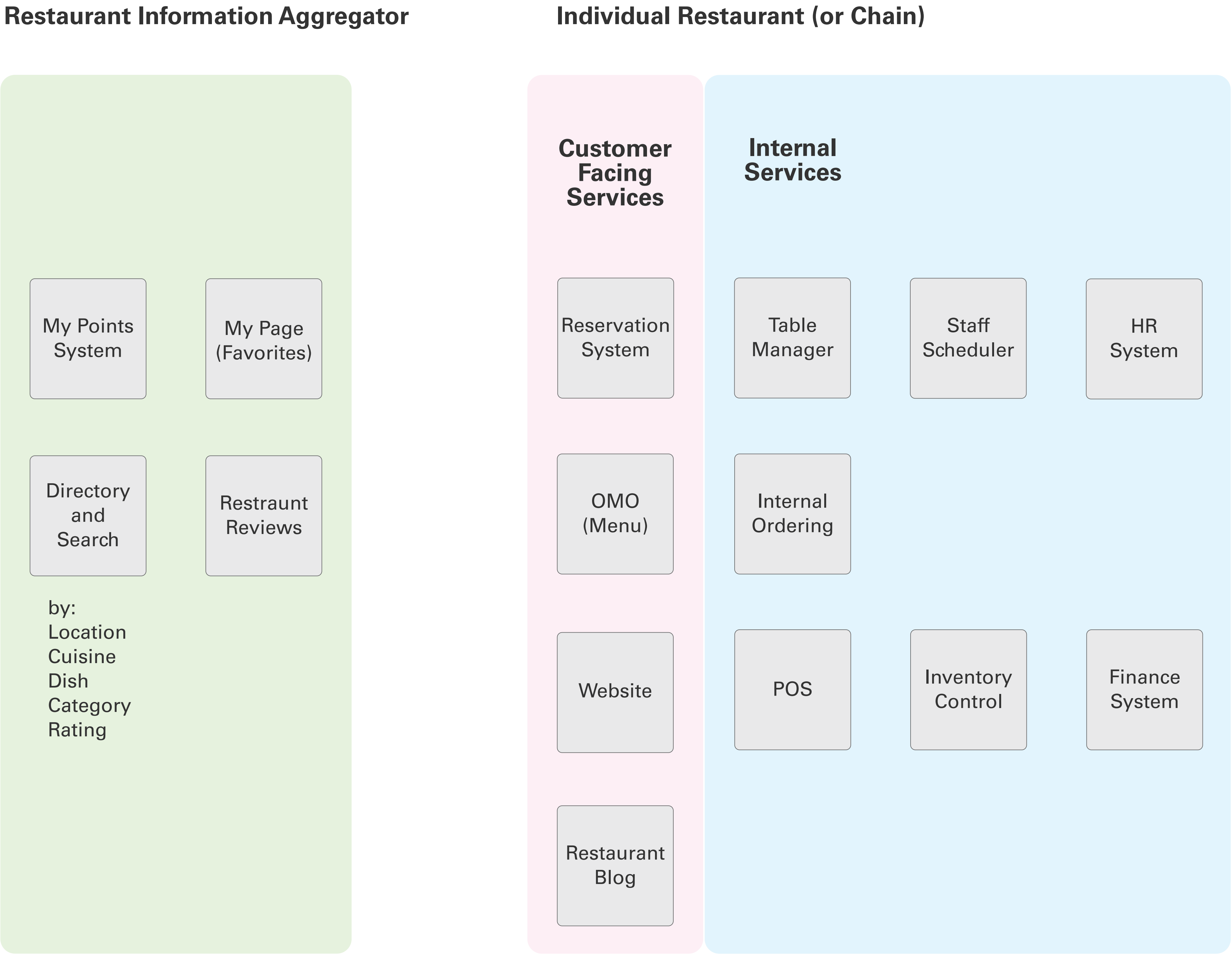
Restaurant: Actor interactions

customer	*reads*	*selects*	*eats*	*pays*	*includes*
	menu;	order;	food;	bill;	tip;
waiter	*brings*	*takes*	*serves*	*presents*	
chef	*creates*		*cooks*		

Restaurant: Basic meal cycle



Restaurant: Information management systems



Restaurant Information Aggregator

My Points System

My Page (Favorites)

Directory and Search

Restraunt Reviews

by:
Location
Cuisine
Dish
Category
Rating

Individual Restaurant (or Chain)

Customer Facing Services

Reservation System

OMO (Menu)

Website

Restaurant Blog

Internal Services

Table Manager

Staff Scheduler

HR System

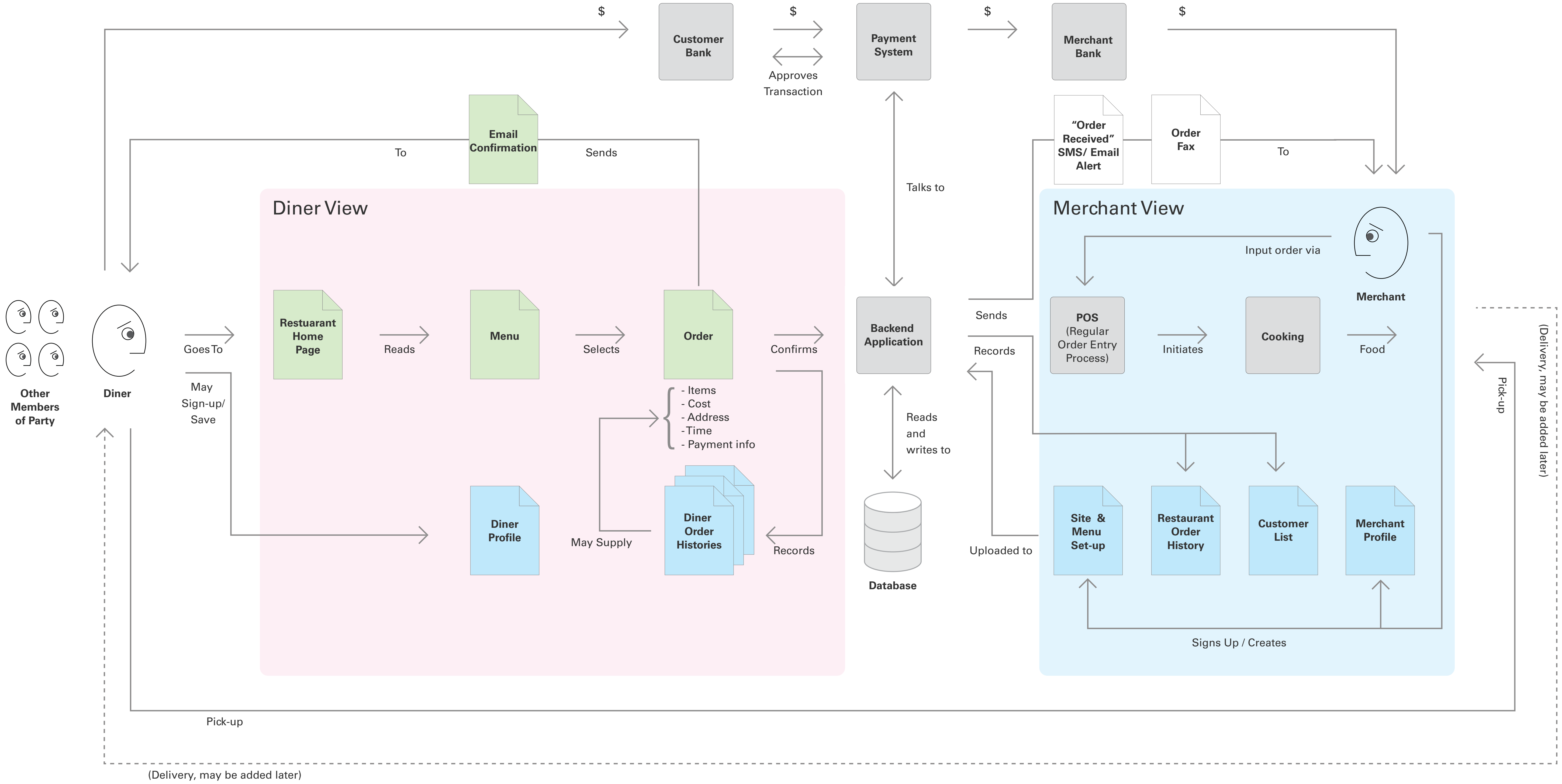
Internal Ordering

POS

Inventory Control

Finance System

Restaurant: Online ordering system: Service Model



$
Customer Bank
$
Approves Transaction
Payment System
$
Merchant Bank
$

To
Email Confirmation
Sends

"Order Received" SMS/ Email Alert
Order Fax
To

Talks to

Diner View

Merchant View
Input order via
Merchant

Other Members of Party
Diner
Goes To
May Sign-up/ Save
Restuarant Home Page
Reads
Menu
Selects
Order
Confirms
- Items
- Cost
- Address
- Time
- Payment info
May Supply
Diner Profile
Diner Order Histories
Records

Backend Application
Reads and writes to
Database

Sends
Records
Uploaded to
POS (Regular Order Entry Process)
Initiates
Cooking
Food
Site & Menu Set-up
Restaurant Order History
Customer List
Merchant Profile
Signs Up / Creates

Pick-up
(Delivery, may be added later)

Pick-up
(Delivery, may be added later)

Rules of thumb

When judging (mental) models, consider four basic criteria

1. **Fit**
   How does the model fit the evidence?
   Is our evidence relevant?
   Is it reliable?
   Is it sufficiently granular? (depth)
   Do we have enough evidence to draw meaningful conclusions?
   Are the elements of the model necessary and sufficient?
   Are the elements of the model "MECE" – mutually exclusive and collectively exhaustive?
2. **Least Means**
   Is there a simpler way to explains the evidence?
   Given two models explaining the same evidence, Ockham told us to prefer the simpler.
3. **Consistency**
   Is the model internally consistent?
   Is it free from contradiction?
4. **Predictive Value**
   What predictions does the model make?
   Are our model's predictions consistent with later observations?
   Do the model's predictions help us make decisions that might have been more difficult without them?

When judging visual representations, consider five primary criteria

1. **Fit**
   Is the representation congruent with the model?
   Do representation and model have similar structures?
   Are all the elements in the model explicit in the representation?
2. **Least Means**
   Could the model be represented in a similar way?
   What can be removed without changing the meaning? (Remove decorations)
   Could conventional symbols or other standard patterns make reading easier?
3. **Consistency**
   Are the means of representation consistent?
   Similar forms should represent similar functions or similar content.
   Likewise, similar functions or similar content should be represented by similar forms.
   Are all elements and their connections labeled?
4. **Contrast**
   What about the model should appear to be most important?
   Does the most important thing appear most important?
   (Not everything is equally important. Important elements of the model should stand out in the representation.
   One way to achieve contrast is through scale, making more important items larger than less important items.)
5. **Hierarchy**
   How do the elements of the system appear to fit together?
   Is the structure of the system clearly visible?
   Do we know where to look first?

The final test of the model (and representation) is with the audience

Does the audience understand it?
Do they agree with it?
Do they agree that they agree?
Will they act on it?

Special thanks to
Anne Chamberlain
Jamie Ikeda

hugh@dubberly.com

Presentation posted at
systems.dubberly.com/models.pdf

Appendix

> *“At its heart, software design is about creating virtual worlds in which users work, learn, and play.*
>
> *Virtuality has two aspects:*
> *1. Conceptual structure—the ideas and how they unfold, connect, and lodge in the mind*
> *2. Feel—how things look and the other sensations we experience*
> *(crude or slick, bumpy or smooth, warm or cool)*
>
> *The real issue is designing a consistent conceptual structure, one that fits the domain as much as possible, as comprehensively and comprehensibly as possible.*
>
> *Consistency, completeness, and clarity are the objectives.”*

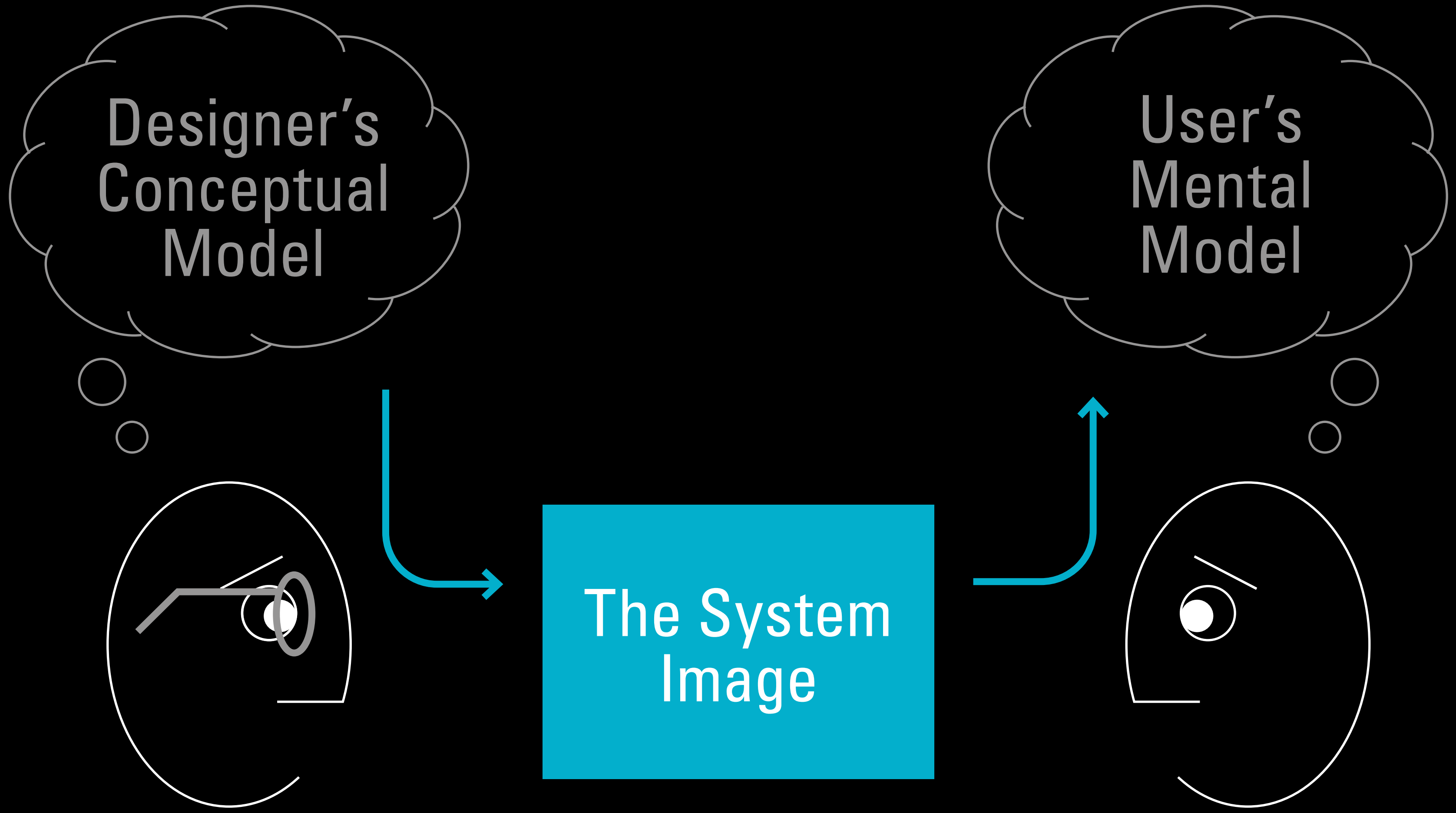
— Ted Nelson, the inventor of hypertext

> *“For people to use a product successfully, they must have the same mental model (the user’s model) as that of the designer (the designer’s model). But the designer only talks to the user via the product itself, so the entire communication must take place through the ‘system image’: the information conveyed by the physical product itself.”*

— Don Norman, *The Design of Everyday Things*, 1988

“...most digital systems fail when they fail to provide a story, when there is a poor conceptual model.”

— Don Norman

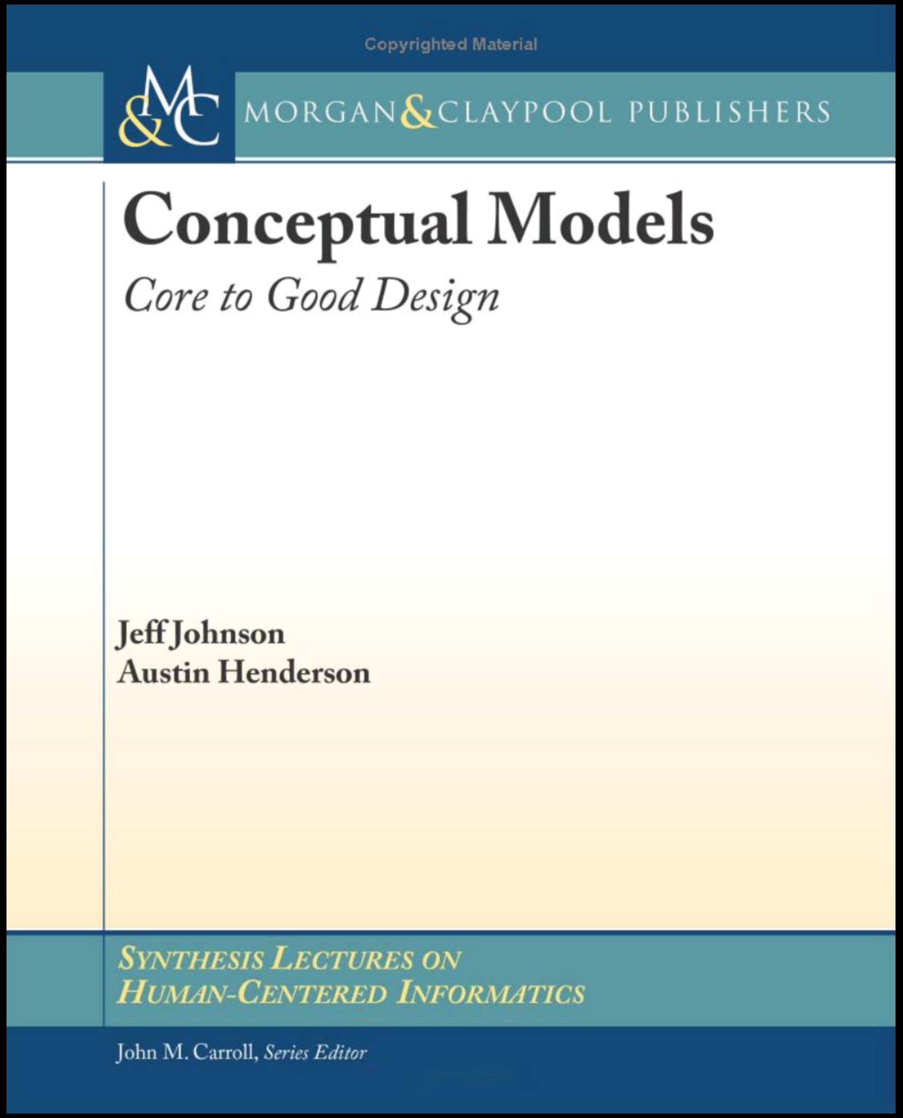


Designer’s Conceptual Model

The System Image

User’s Mental Model

How to make conceptual models is explained in a wonderful new book *Conceptual Models: Core to Good Design,* by Johnson & Henderson



Copyrighted Material

MORGAN & CLAYPOOL PUBLISHERS

Conceptual Models

Core to Good Design

Jeff Johnson
Austin Henderson

SYNTHESIS LECTURES ON HUMAN-CENTERED INFORMATICS

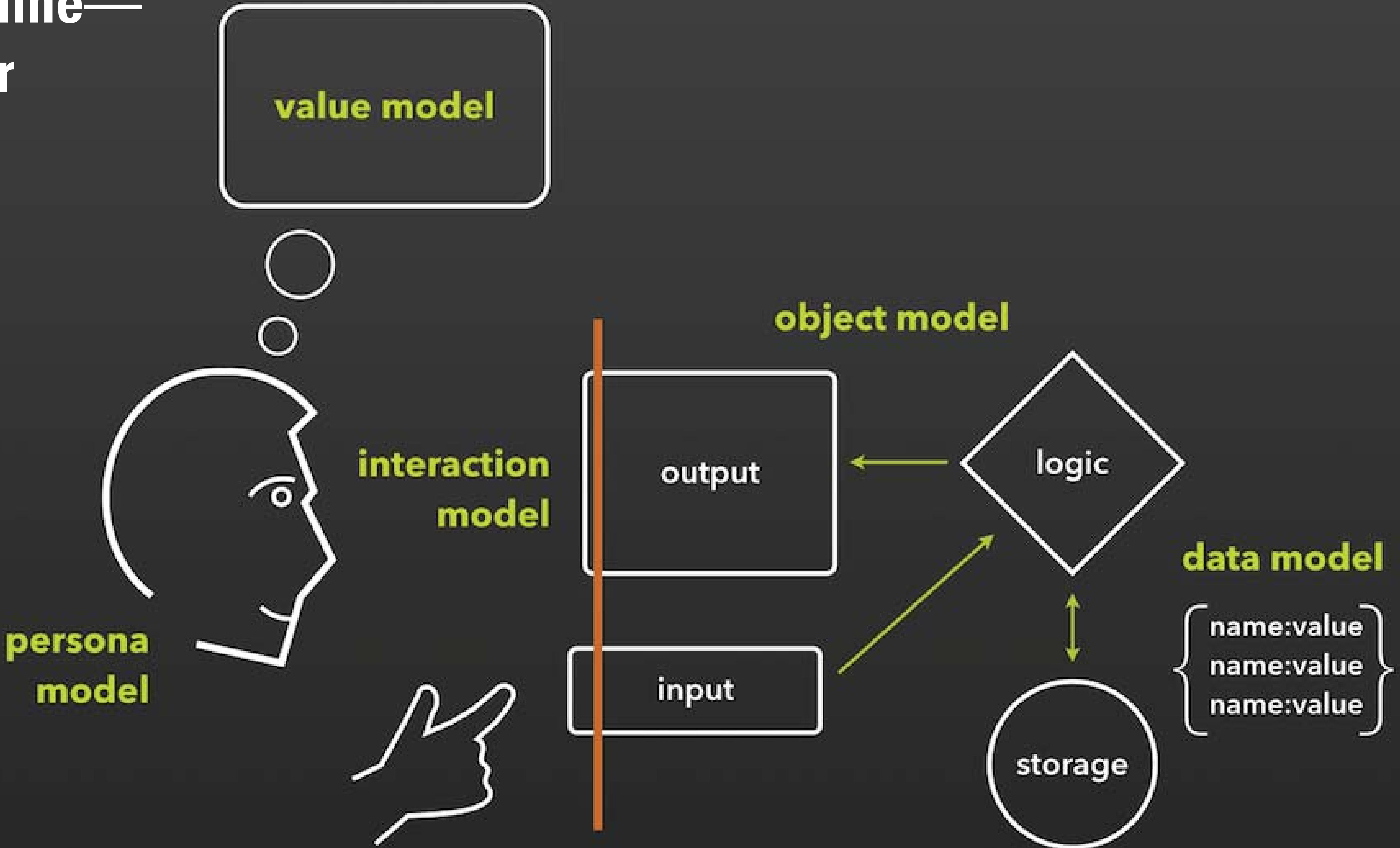
John M. Carroll, *Series Editor*

A conceptual model describes what a user needs to know in order to use your application successfully.

> *"A conceptual model is a high-level description of an application. It enumerates all concepts in the application that users can encounter, describes how those concepts relate to each other, and how those concepts fit into tasks that users perform with the application."*

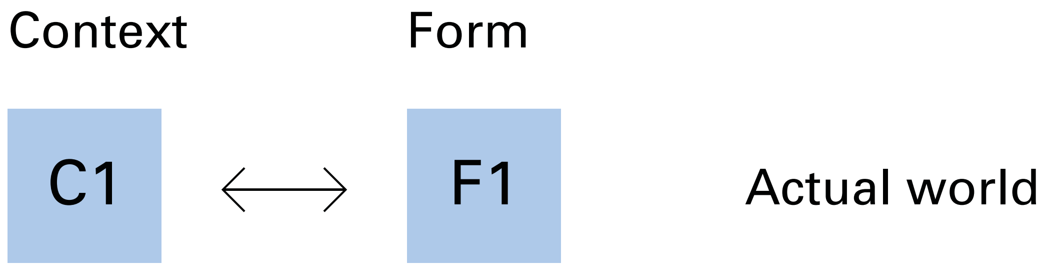
— Jeff Johnson + Austin Henderson, *Conceptual Models: Core to Good Design*, 2012

Digital Machine—
Tim Scheiner



Before the industrial revolution, craftsmen designed things as they made them.

1. Un-self-conscious Design



Context

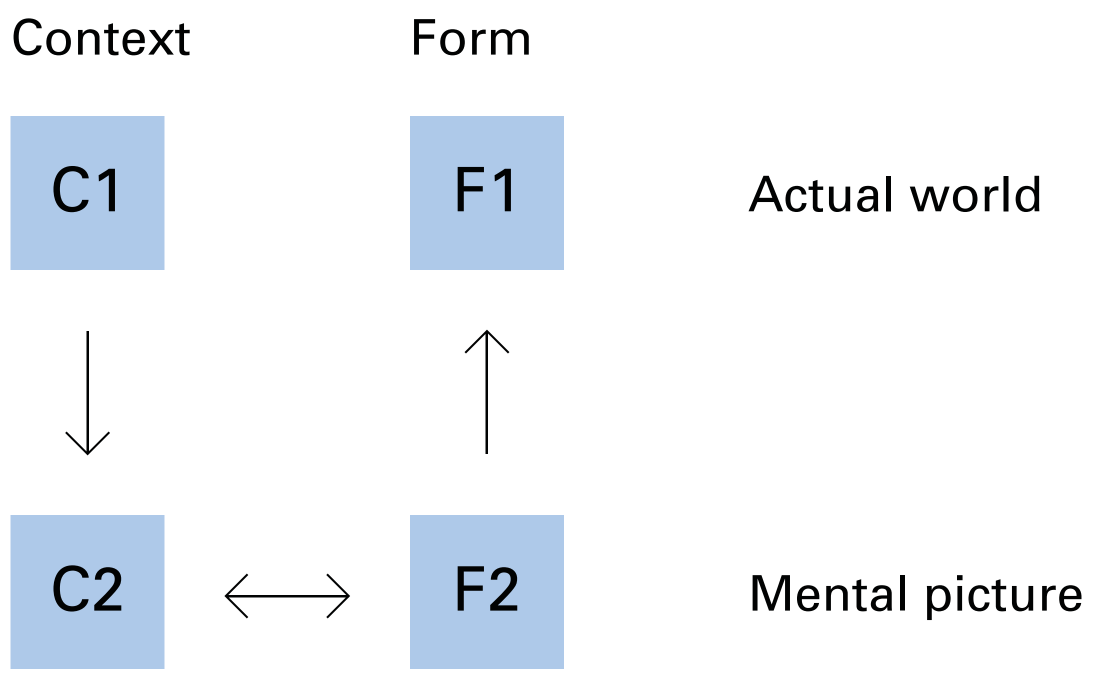
Form

C1 ⟷ F1

Actual world

During the industrial revolution, designing became separated from making.

2. Self-conscious Design



Context

Form

C1

F1

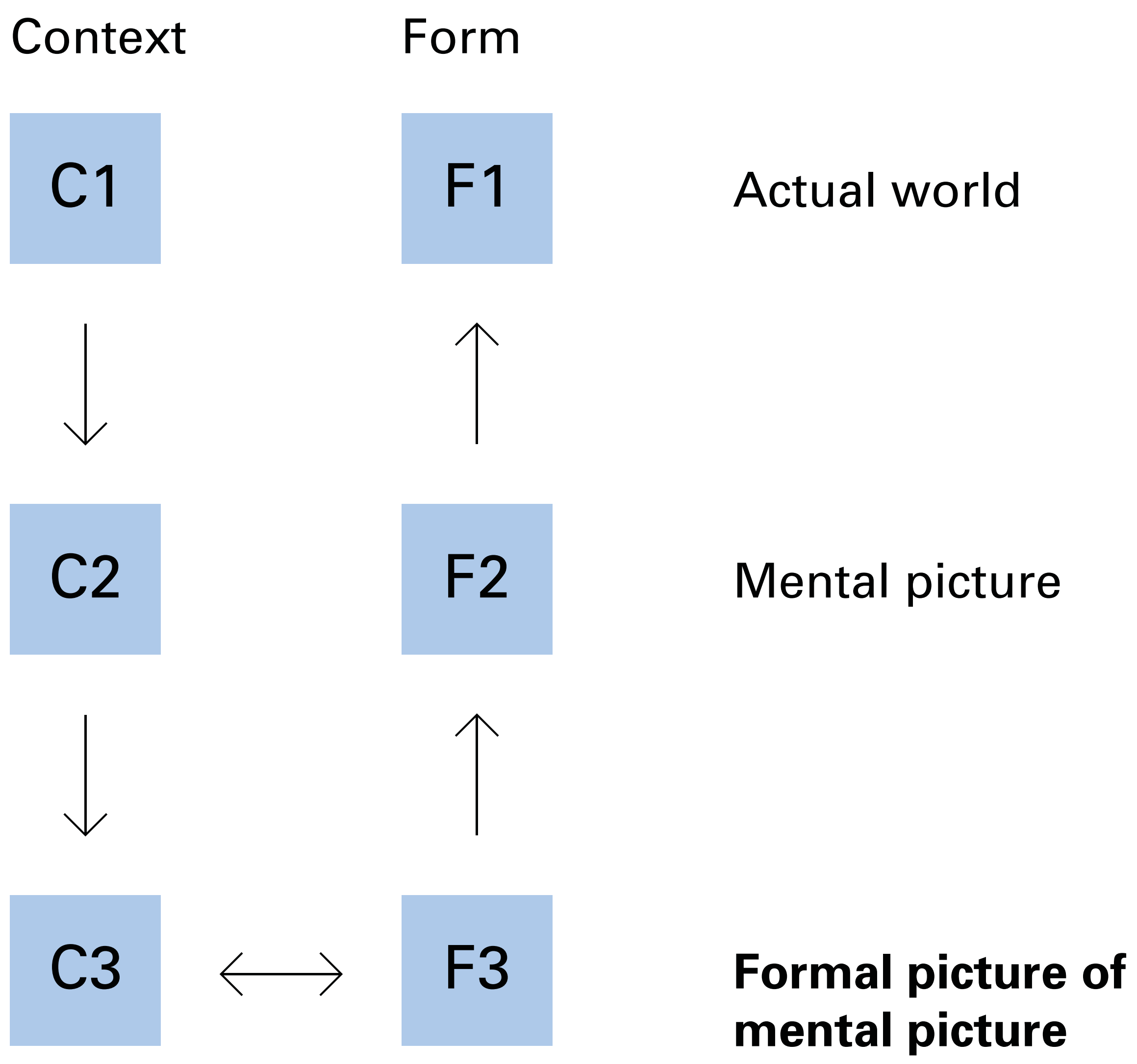
Actual world

C2 ⟷ F2

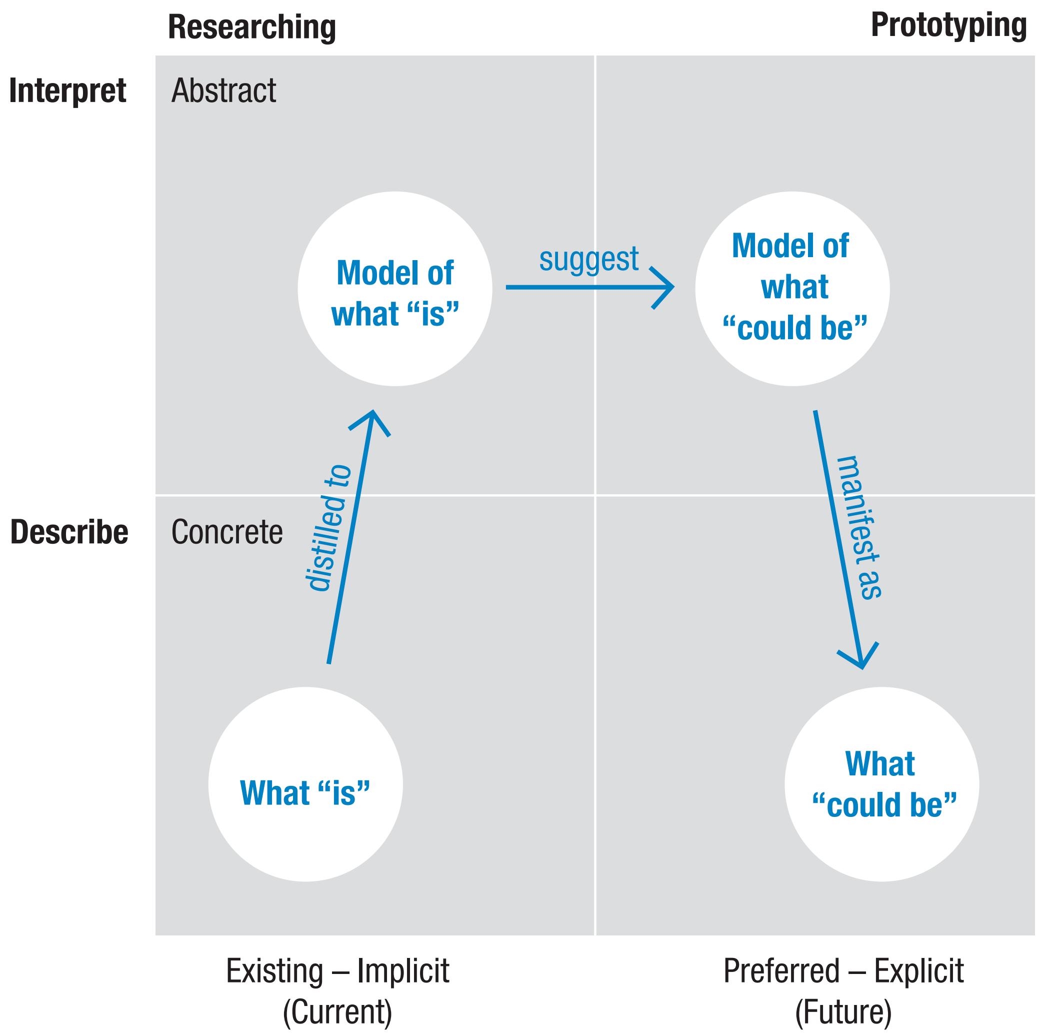
Mental picture

Designers began to rely increasingly on drawing.

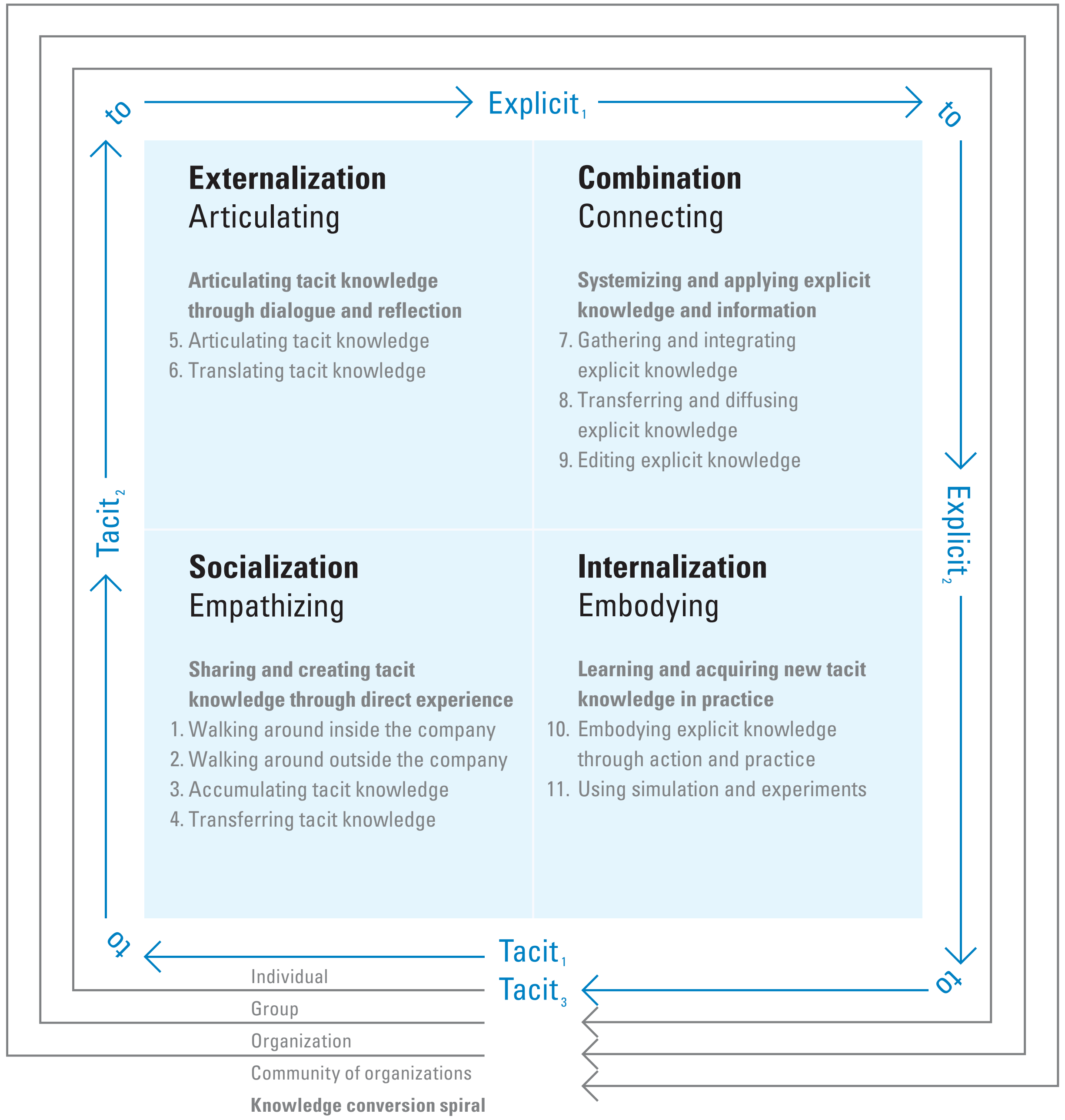
3. Mediated Design


Context
Form
C1
F1
Actual world
C2
F2
Mental picture
C3
F3
Formal picture of mental picture

The Analysis-Synthesis Bridge Model shows how design crosses the gap between *what is* and *what should be*.



The **SECI Model** shows how organizations turn tacit knowledge into explicit knowledge, create new knowledge, and deploy it in operations.



to Explicit$_1$ to

Tacit$_2$

Explicit$_2$

Externalization
Articulating

Articulating tacit knowledge through dialogue and reflection

5. Articulating tacit knowledge
6. Translating tacit knowledge

Combination
Connecting

Systemizing and applying explicit knowledge and information

7. Gathering and integrating explicit knowledge
8. Transferring and diffusing explicit knowledge
9. Editing explicit knowledge

Socialization
Empathizing

Sharing and creating tacit knowledge through direct experience

1. Walking around inside the company
2. Walking around outside the company
3. Accumulating tacit knowledge
4. Transferring tacit knowledge

Internalization
Embodying

Learning and acquiring new tacit knowledge in practice

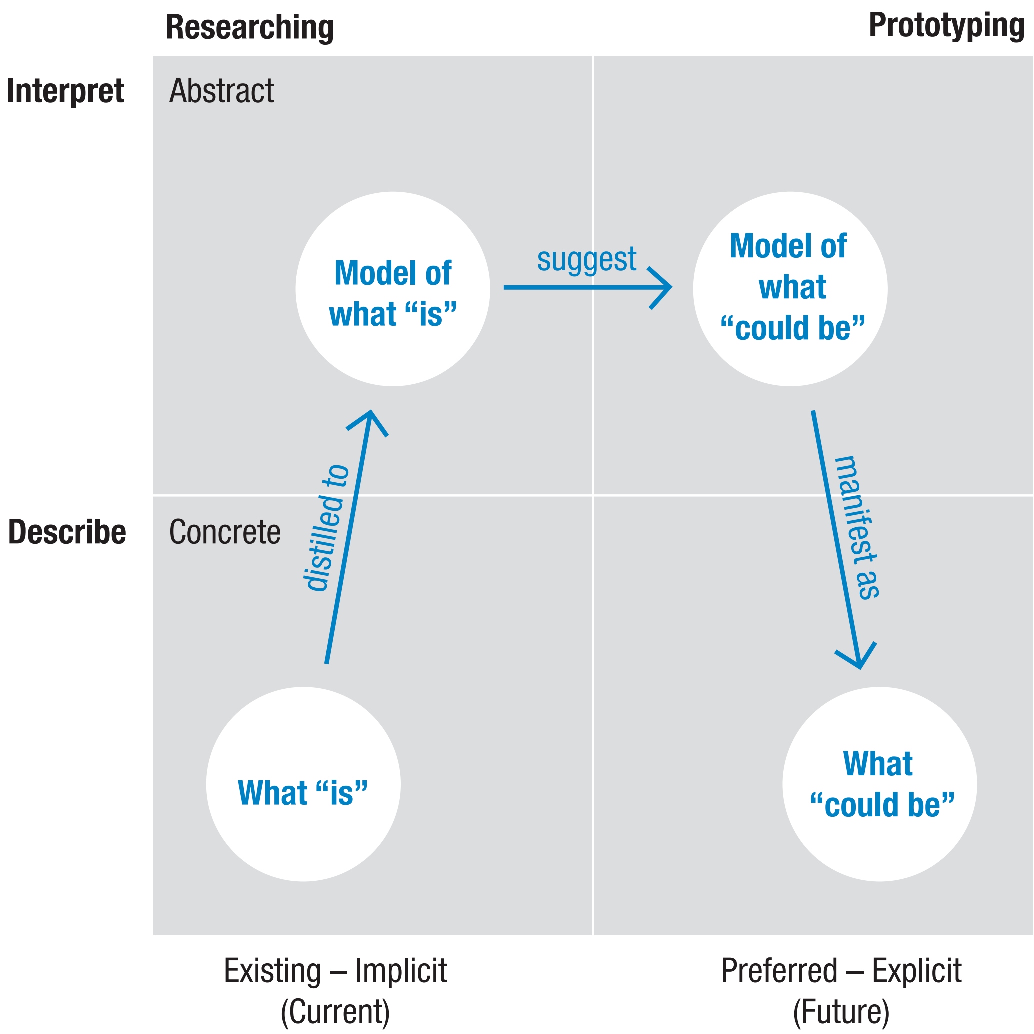
10. Embodying explicit knowledge through action and practice
11. Using simulation and experiments

to Tacit$_1$ to

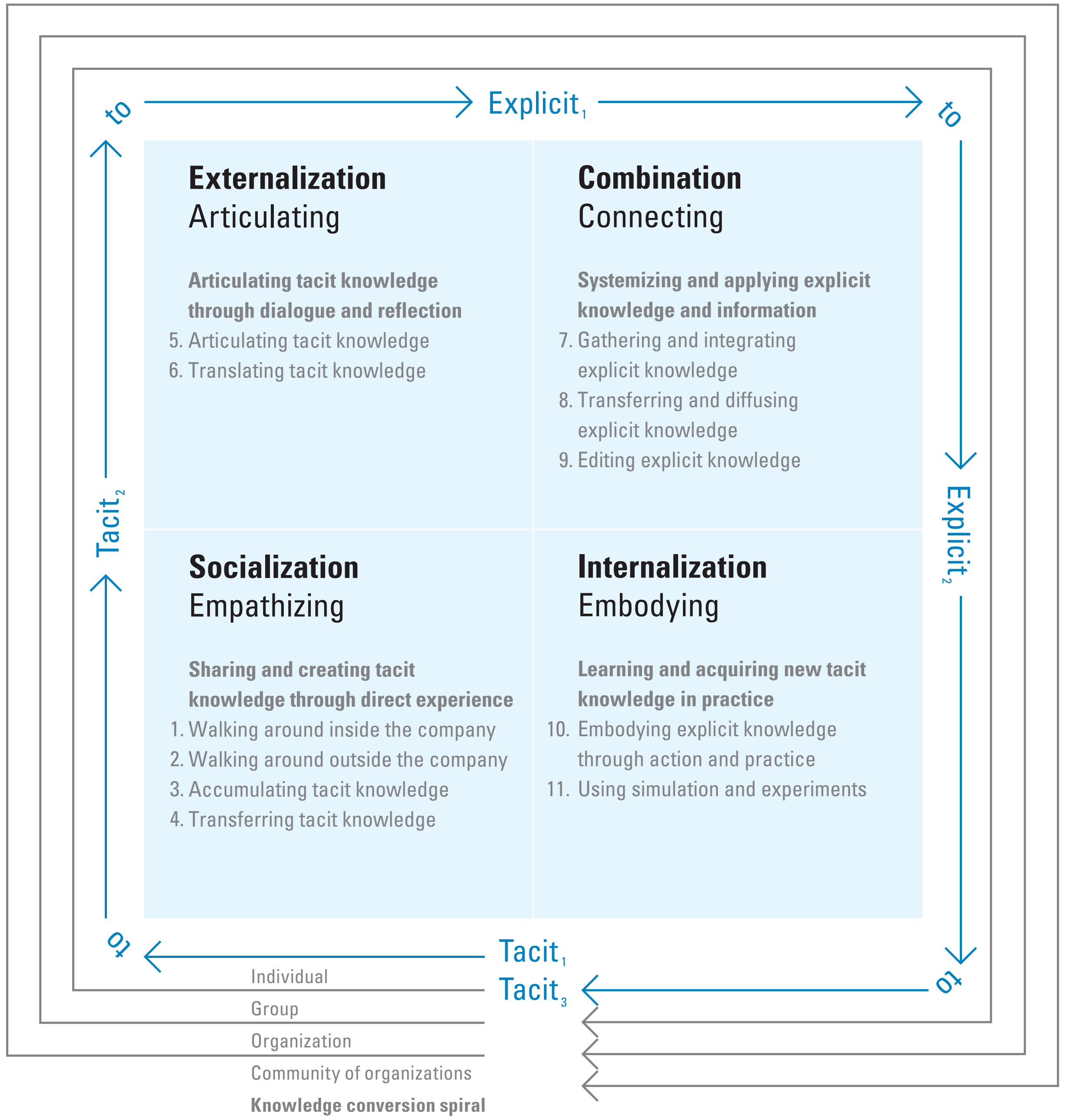
Tacit$_3$

Individual
Group
Organization
Community of organizations
Knowledge conversion spiral

Both models have the same basic structure—iterative loops—suggesting that designing *is* learning.



Analysis-Synthesis Bridge Model
Dubberly, Evenson & Robison (2008)



Externalization
Articulating

Articulating tacit knowledge through dialogue and reflection

5. Articulating tacit knowledge
6. Translating tacit knowledge

Combination
Connecting

Systemizing and applying explicit knowledge and information

7. Gathering and integrating explicit knowledge
8. Transferring and diffusing explicit knowledge
9. Editing explicit knowledge

Socialization
Empathizing

Sharing and creating tacit knowledge through direct experience

1. Walking around inside the company
2. Walking around outside the company
3. Accumulating tacit knowledge
4. Transferring tacit knowledge

Internalization
Embodying

Learning and acquiring new tacit knowledge in practice

10. Embodying explicit knowledge through action and practice
11. Using simulation and experiments

SECI model of knowledge create
Ikujiro Nonaka (1995)