

Google
Experience Talk

Using Conceptual Models in Interaction Design

June 6, 2012

Hugh Dubberly
Dubberly Design Office

**When you say,
“I know how to use Photoshop,”
or “I know how to use Excel,”
you refer to a conceptual model.**

**Knowing how to use an application—
or any tool—
means having a conceptual model
of the application.**

**At their core,
all applications have a main data type,
and tools for selecting and changing data.**

**If you understand the main data type
and the tools for selecting and changing data,
then you understand the application—
or at least much of what it can do.**

What defines a spreadsheet?

Spreadsheets are **grids of cells**,
containing **text, numbers, and functions**
that operate on a range of cells

This is what makes VisiCalc similar to Excel.

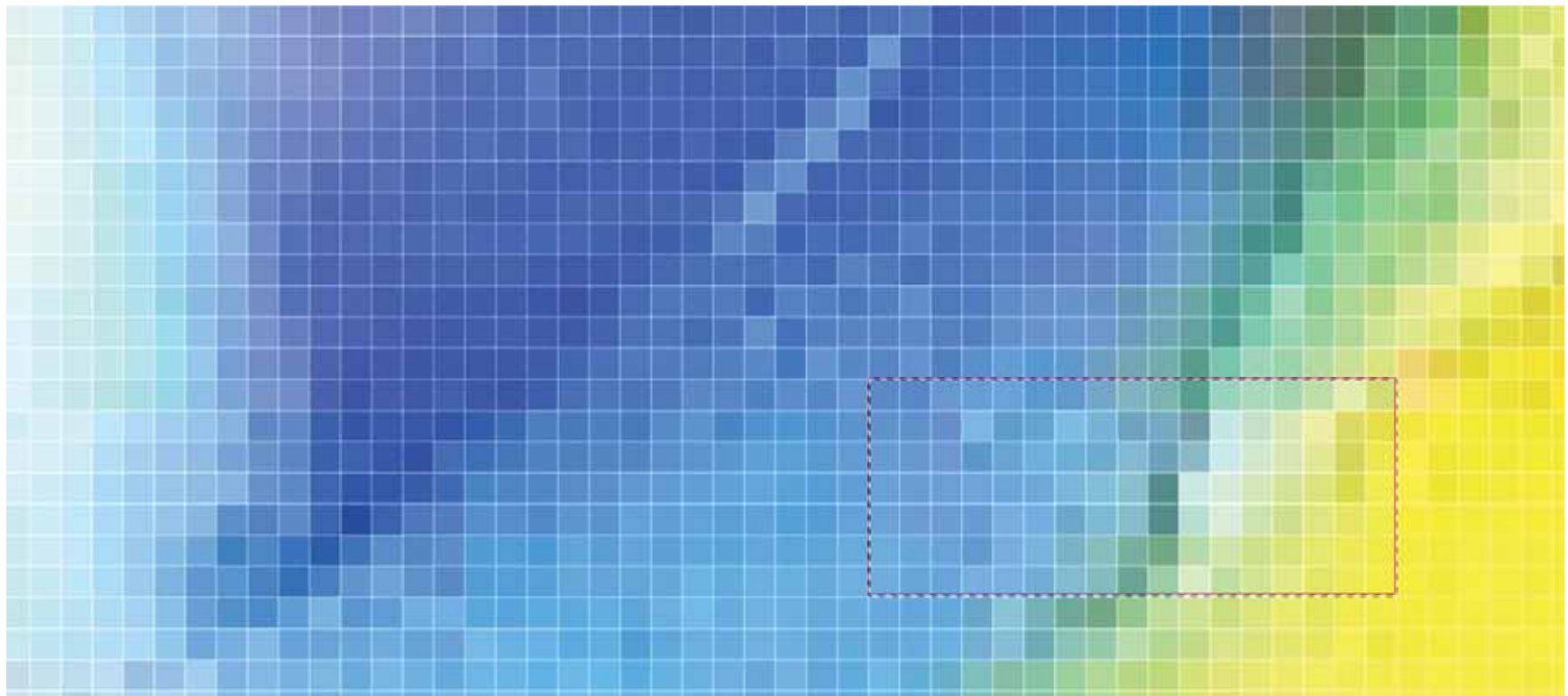
◇	A	B	C	D
1		FY 10	FY 11	
2	Jan	1	2	
3	Feb	3	4	
4	March	5	6	
5	April	7	8	
6				
7	Totals	16	=SUM(C2:C5)	
8				
9				
10				

What defines an image editor?

Images are **grids of cells**,
containing **numbers (that represent colors)**.

You edit an image by selecting a range of cells
and applying a transform function.

This is what makes **MacPaint** similar to **Photoshop**.



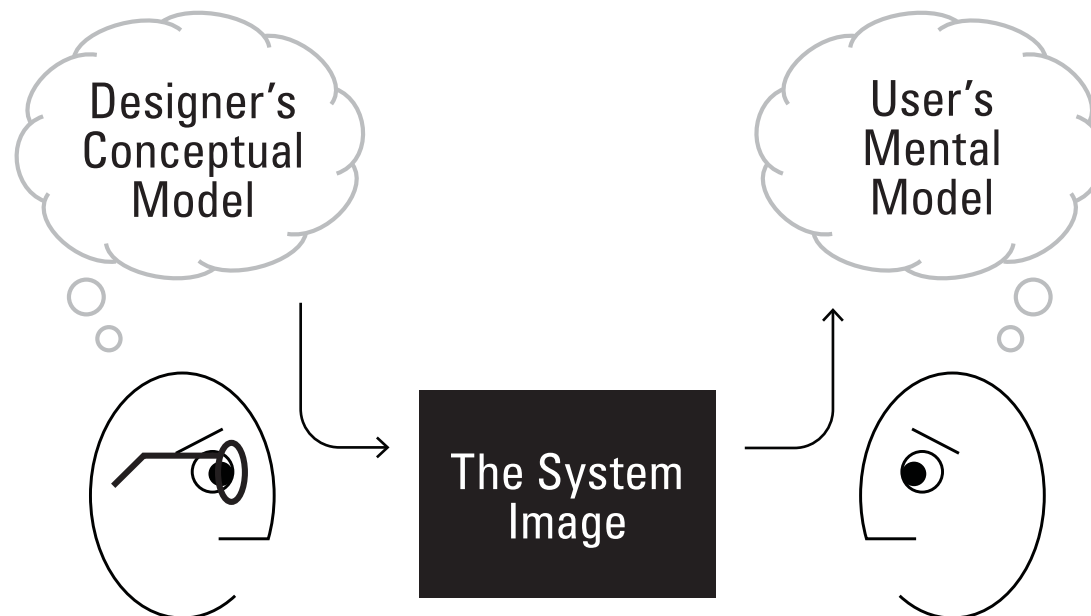
I believe
that understanding the conceptual model
is key to interaction design;

that's because
users need to understand the model
in order to be able to use a product easily.

Don Norman talks about a similar idea: The system image

“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



In addition to communicating the conceptual model to users, designers also need to communicate it to other designers—
and to product managers and senior management, and, of course, to engineers.

And the communication needs to go **two ways;
it needs to be a conversation.**

The idea of a conversation about conceptual models of a product or service raises some questions.

How might we design these conversations?

How might we make them rigorous?

What sorts of artifacts might support them?

Is there a necessary and sufficient set of artifacts?

Foundation

Steve Job's definition of design

“In most people’s vocabularies, design means veneer. It’s interior decorating. It’s the fabric of the curtains and the sofa. But to me, nothing could be further from the meaning of design. Design is the fundamental soul of a man-made creation that ends up expressing itself in successive outer layers of the product or service.”

—Steve Jobs, *Fortune*, January 24, 2000

Steve Job's definition of design

*“In most people’s vocabularies, design means veneer. It’s interior decorating. It’s the fabric of the curtains and the sofa. But to me, nothing could be further from the meaning of design. Design is the **fundamental soul** of a man-made creation that ends up expressing itself in **successive outer layers of the product or service.**”*

—Steve Jobs, *Fortune*, January 24, 2000

Should we take Steve Jobs literally?
Is it worth asking:

What's a product's **fundamental soul?**

What are the **successive outer layers?**

Let's start with a model of the layers. Perhaps simplest: 2 layers

- **Function**

- **Form**

—Louis Sullivan, *The Tall Office Building Artistically Considered*, 1896

Linguistics and Semiotics (the theory of signs) suggest 3 layers

- **Pragmatic** or Context or Why are we doing this?
- **Semantic** or Meaning or What are we doing?
- **Syntactic** or Grammar or How are we doing it?

—Charles W. Morris, *Foundation of the Theory of Signs*, 1938

Looking at components may suggest 4 or even more layers

- **System:** a community or marketplace of products interacting
e.g., the Kindle, reader application, Whisper-net network, and Amazon store eco-system
- **Product:** a series of connected and related screens
e.g., the iPhone as a new paradigm in touch-screen-based mobile computing UIs
- **Screen:** a collection of interaction components
e.g., the Windows Phone 7 home screen
- **Widget:** an interaction component, often repeated
e.g., one-button check-out or fan-able pull-down menus

Planning roles may suggest 5 layers

- **Strategy:** User needs + site objectives
- **Scope:** Functional + content specs
- **Structure:** Information design (interface + navigation)
- **Skeleton:** Interaction design + Information architecture
- **Surface:** Visual design

—Jesse Garret, *The Elements of User Experience*, 2003

**And we could continue to divide more finely,
or we could slice along different dimensions.**

**But the main point is
that the layers depend on each other.
And to a large extent—at least in great products—
the successive outer layers derive from a core,
from a fundamental soul.**

**Most UX design
takes place at the outermost layers,
at the screen level—
by iterating on wireframes
or Photoshop mock-ups.**

**Iteration at the screen level is necessary,
but sometimes a focus on details
can hide the big picture;
it can hide the fundamental soul of the product.**

Historically,
design hasn't paid much attention to concepts,
to methods for defining the fundamental soul
of a product.

We have a few: elevator pitch, story treatment,
competitive positioning, value proposition,
unique selling proposition, solution space models,—
and now conceptual models.

Definitions

It may be easier to remember a simpler model, nicely parallel to the Charles W. Morris model

- **Concepts**

e.g., all concepts that the application's user interface exposes to users

- **Task flow**

e.g., the sequence of operations that users execute to accomplish tasks

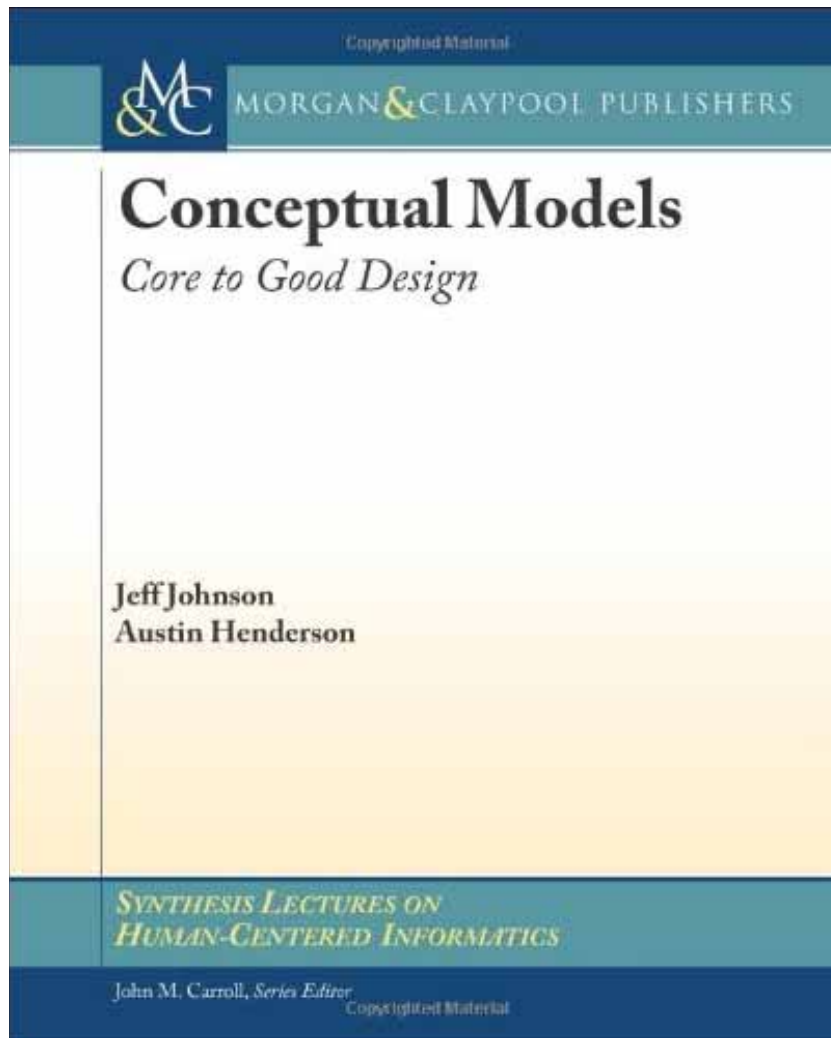
- **Presentation**

e.g., the controls, displays, etc. that comprise its user interface

*“... interaction design consists of concepts, task flow,
and presentation.”*

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

From a wonderful new book



“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 concept fit into tasks that users perform with the application.”

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

**More precisely,
a conceptual model of
an interactive application includes,
the data “objects,” which a user may encounter,
and the “operations, attributes, and relationships,”
which a user may perform on the data objects.**

**It's important to note
that **the conceptual model is independent**
of task flow and presentation.**

**As you work out task flow,
you may find problems in the conceptual model,
and you may need to modify it.**

As an example of a conceptual model Johnson + Henderson describe an alarm clock.

The clock **stores** the **current time** of day,
continually **updating** it to track the passage of time.

It **displays** the current time constantly.

Users can **set** the current time.

Users can set an **alarm** at a **specified time**, or no alarm.

When an alarm is set and the current time equals the set **alarm time**,
the **alarm** is triggered.

Users can **turn off** an alarm.

Johnson + Henderson's conceptual model of a clock could be expressed in digital or analog travel clocks.



Their conceptual model of a clock could also describe the iPhone and Galaxy Nexus clocks.



A critique and a proposal

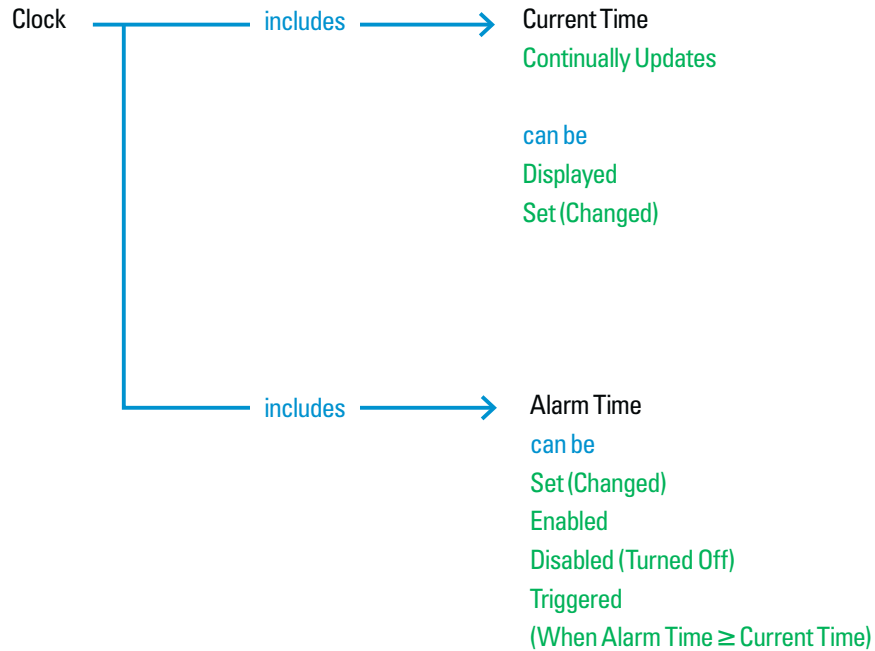
Lists are not models.

**Sentences are not efficient means
of describing structure.**

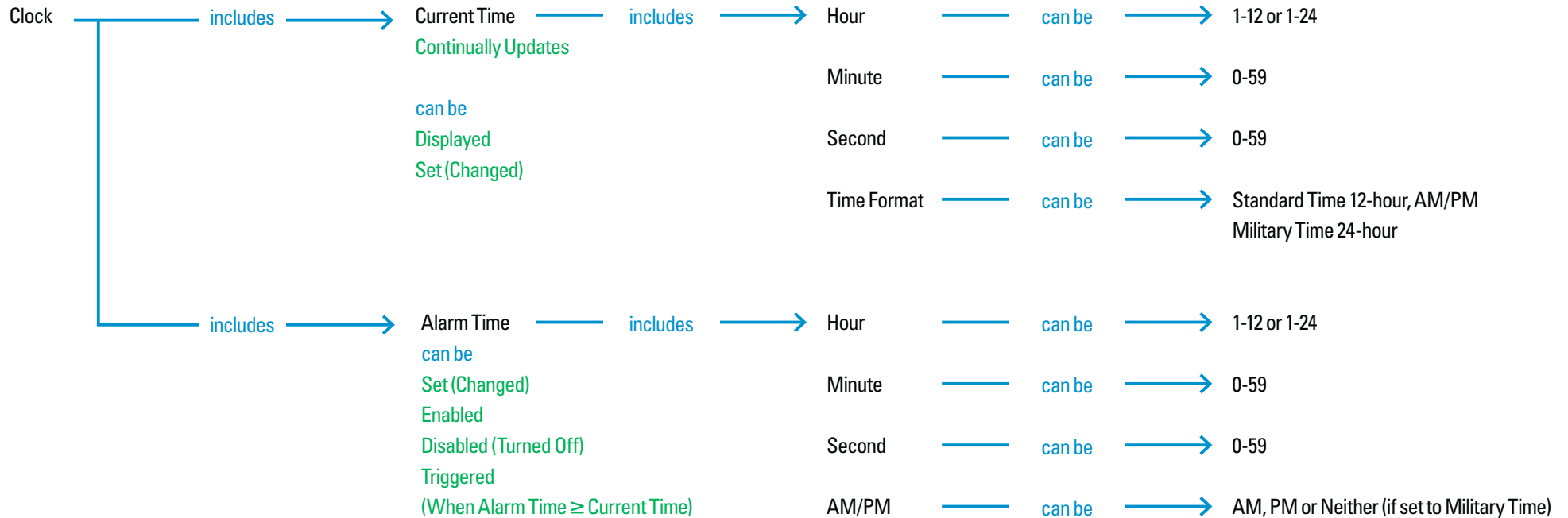
Text is often redundant.

**Node-link diagrams—concept maps—
are a more efficient means
of representing conceptual models**

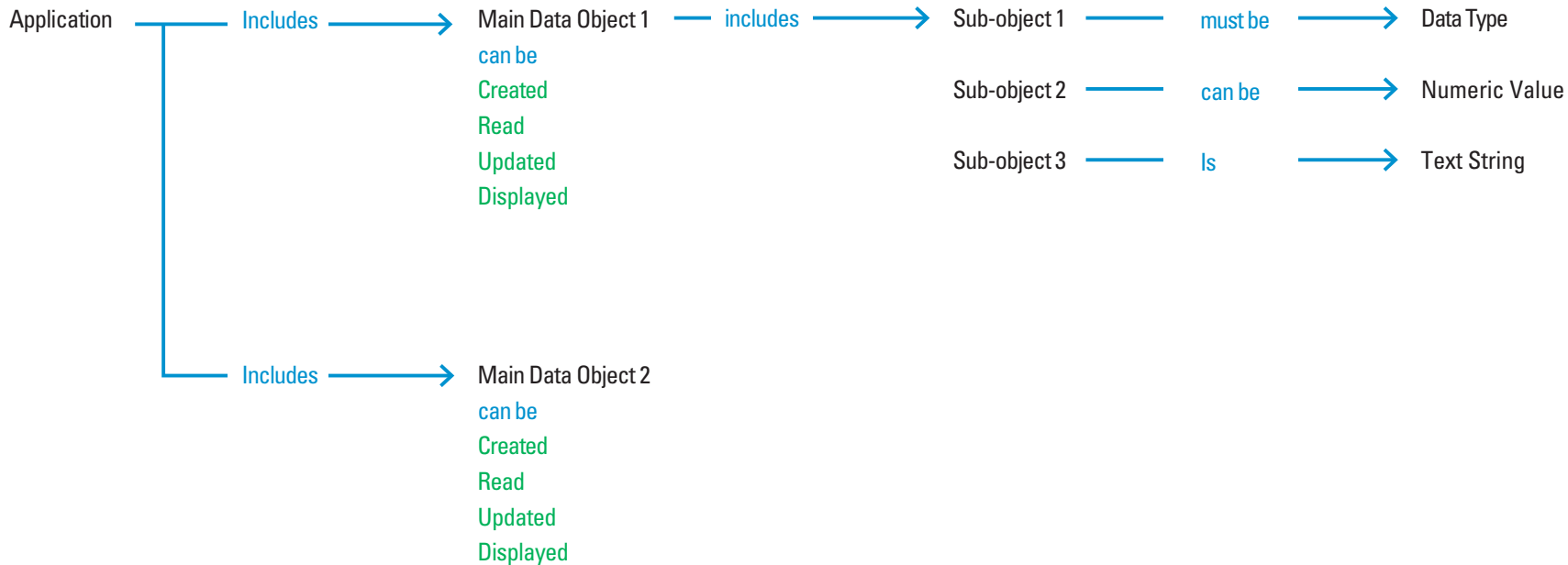
A conceptual model of an alarm clock represented as a concept map.



Conceptual model of an alarm clock with detail added.

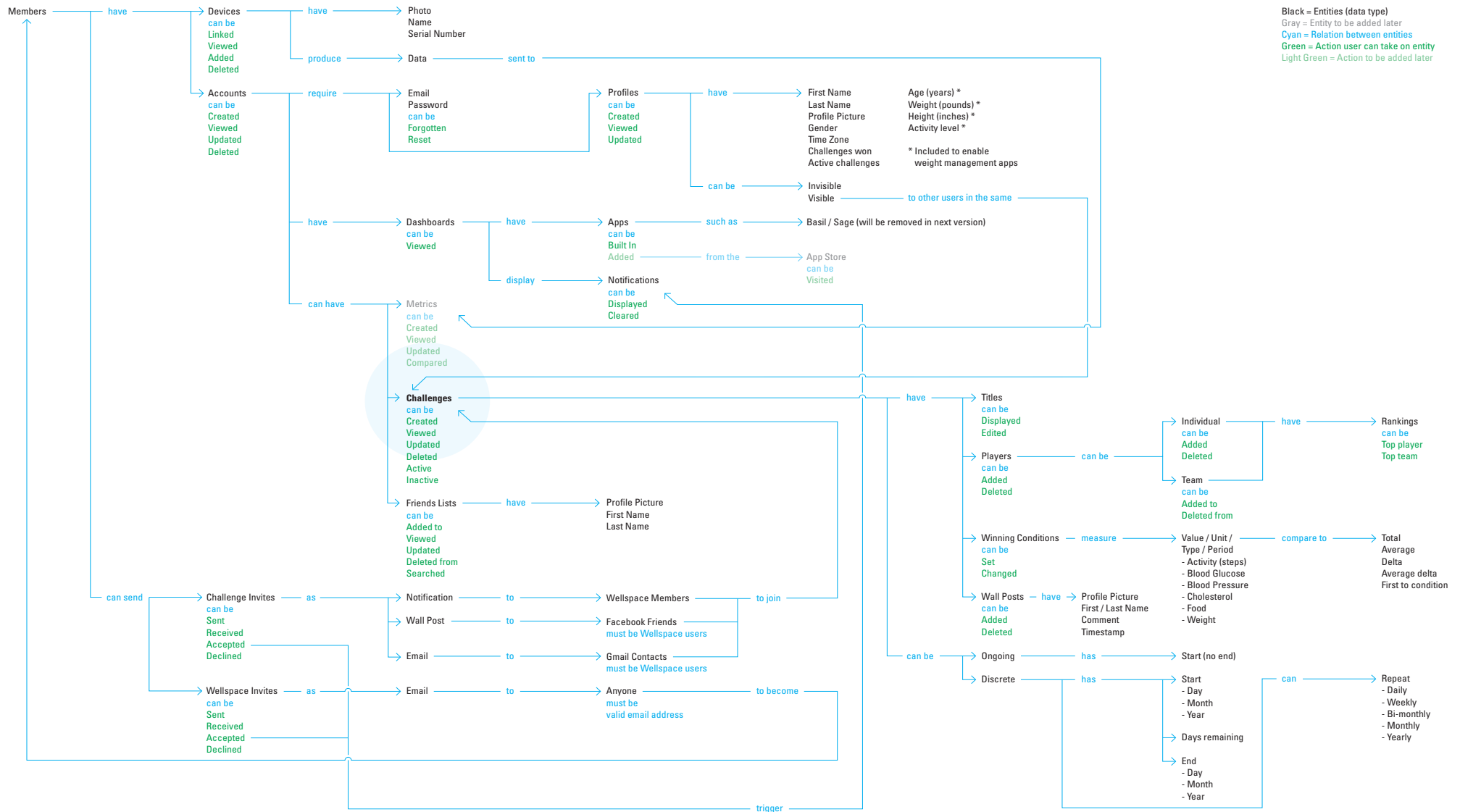


We might formalize the representation.



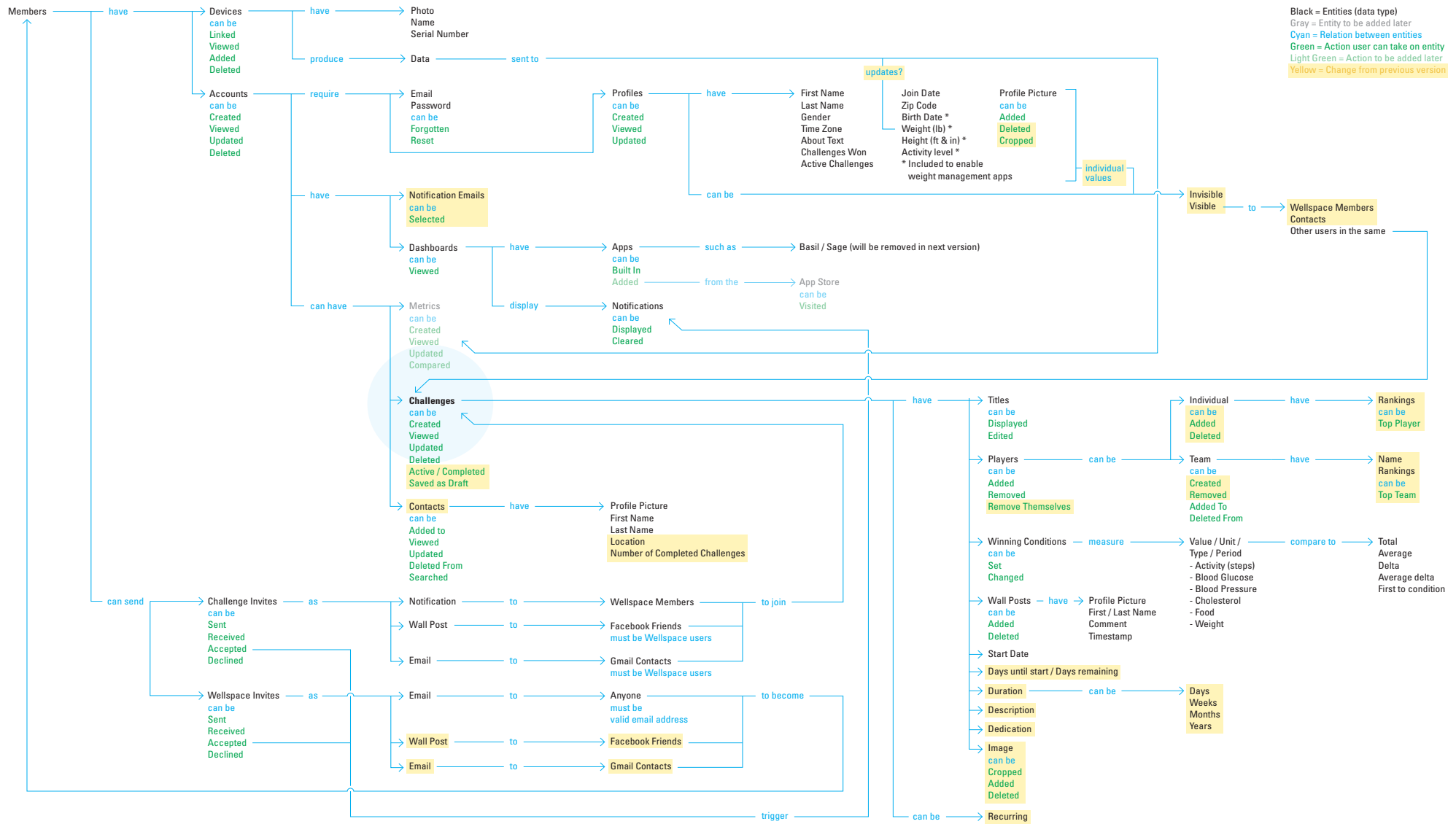
Black = data objects
Blue = relationships
Arrows = data structure
Green = operations
Tinted colors = future

Health challenge example



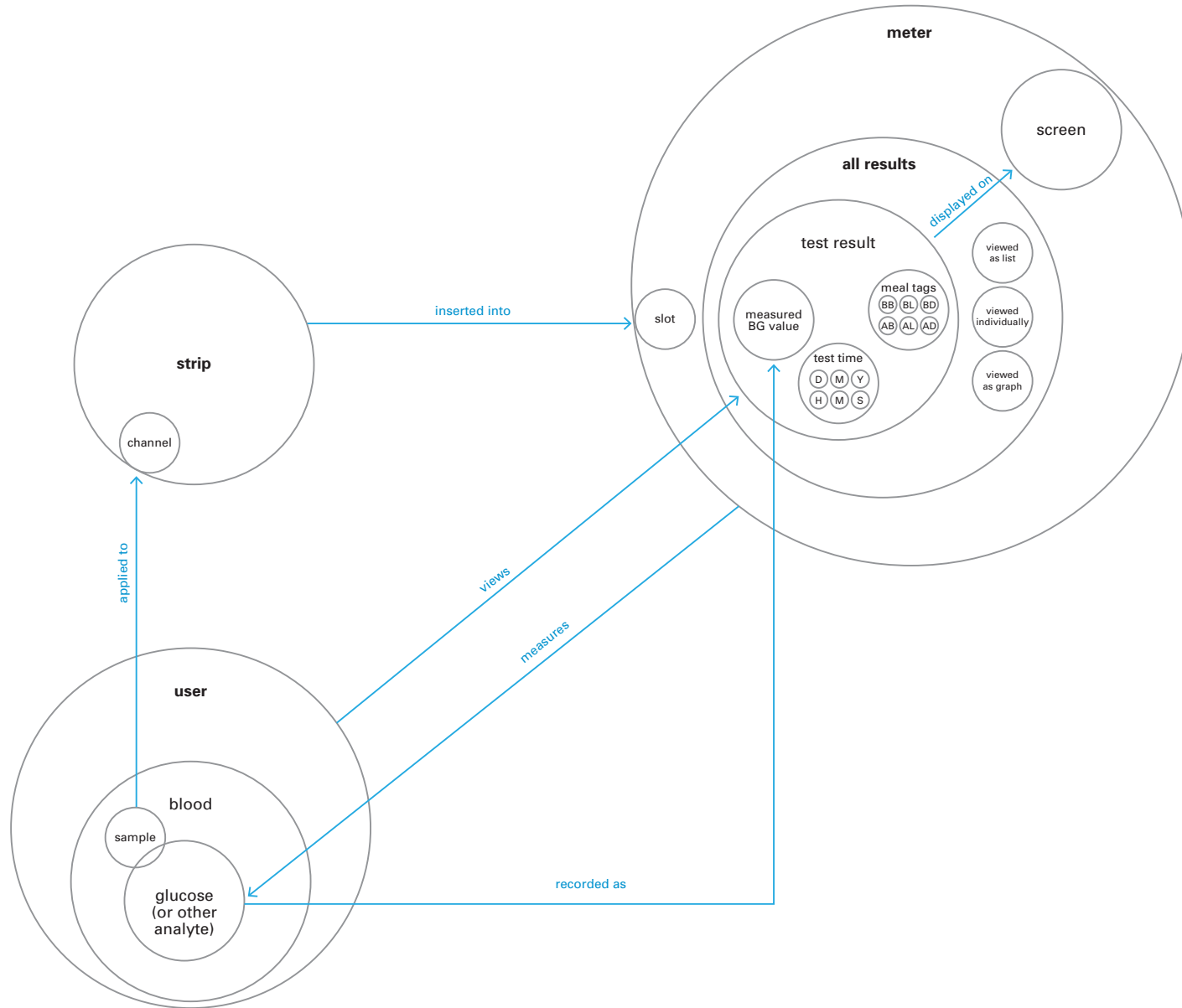
Black = Entities (data type)
 Gray = Entity to be added later
 Cyan = Relation between entities
 Green = Action user can take on entity
 Light Green = Action to be added later

Changes after wireframe design

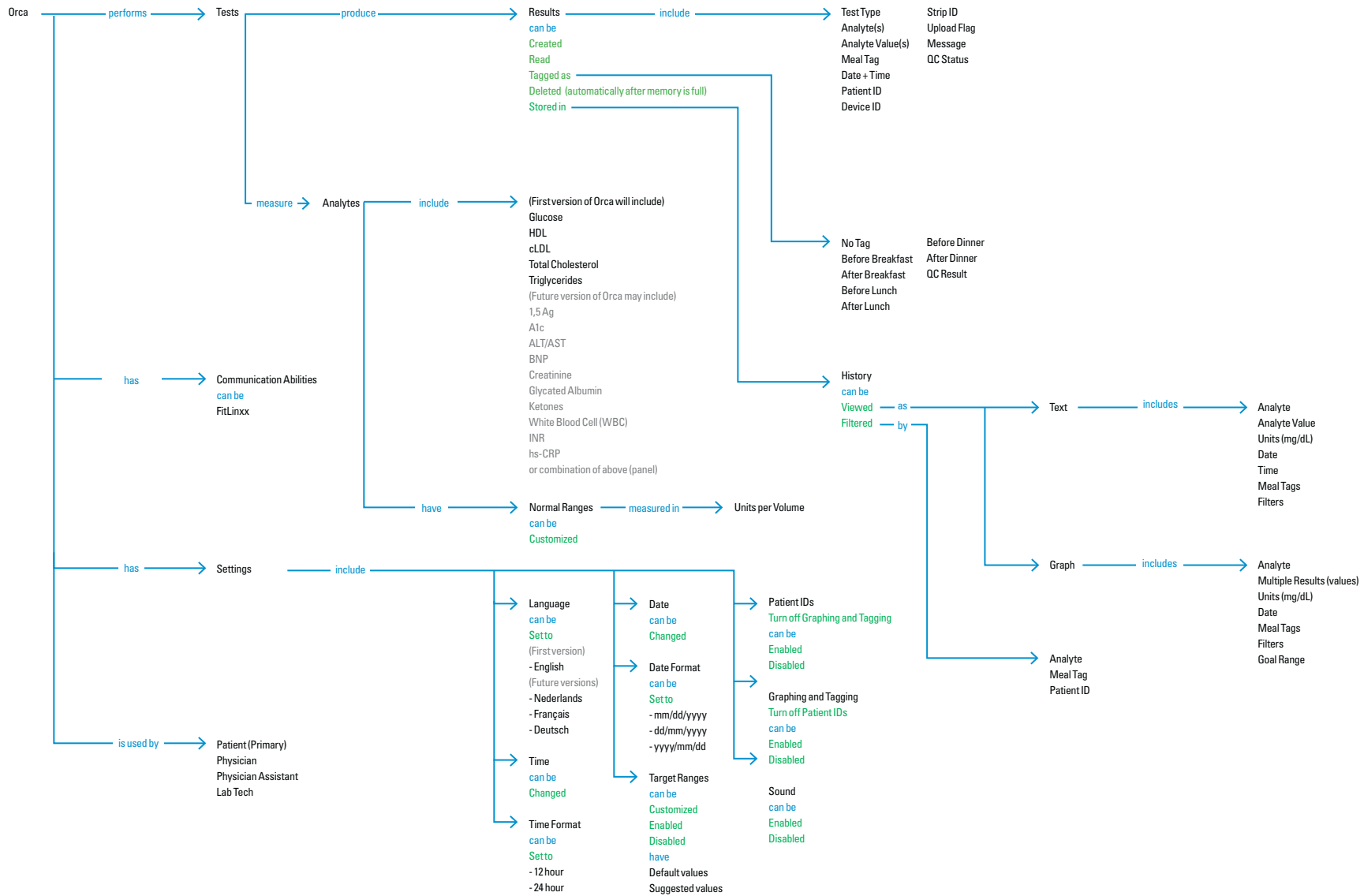


Case study

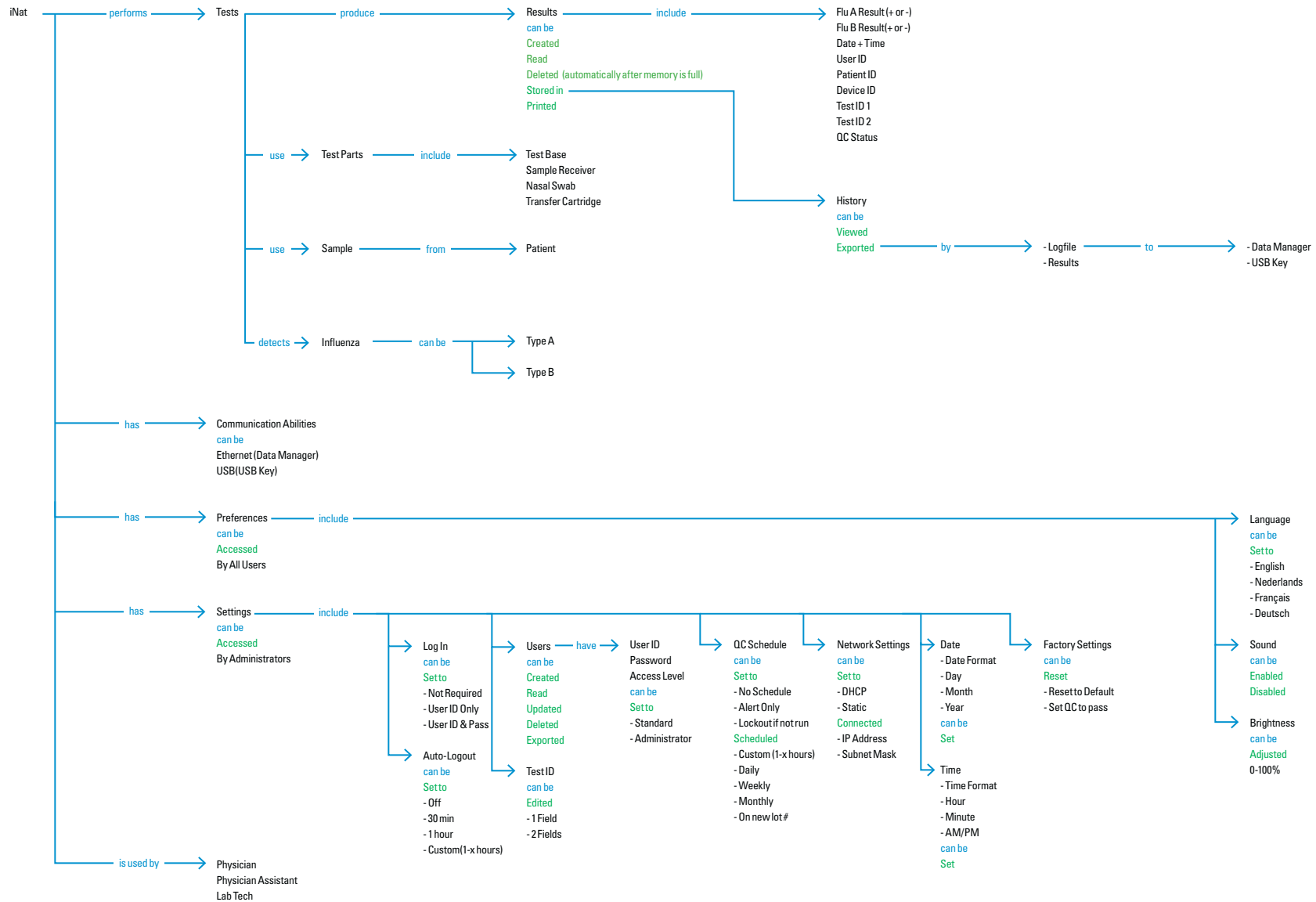
Orca: Blood Analyte Meter, early version



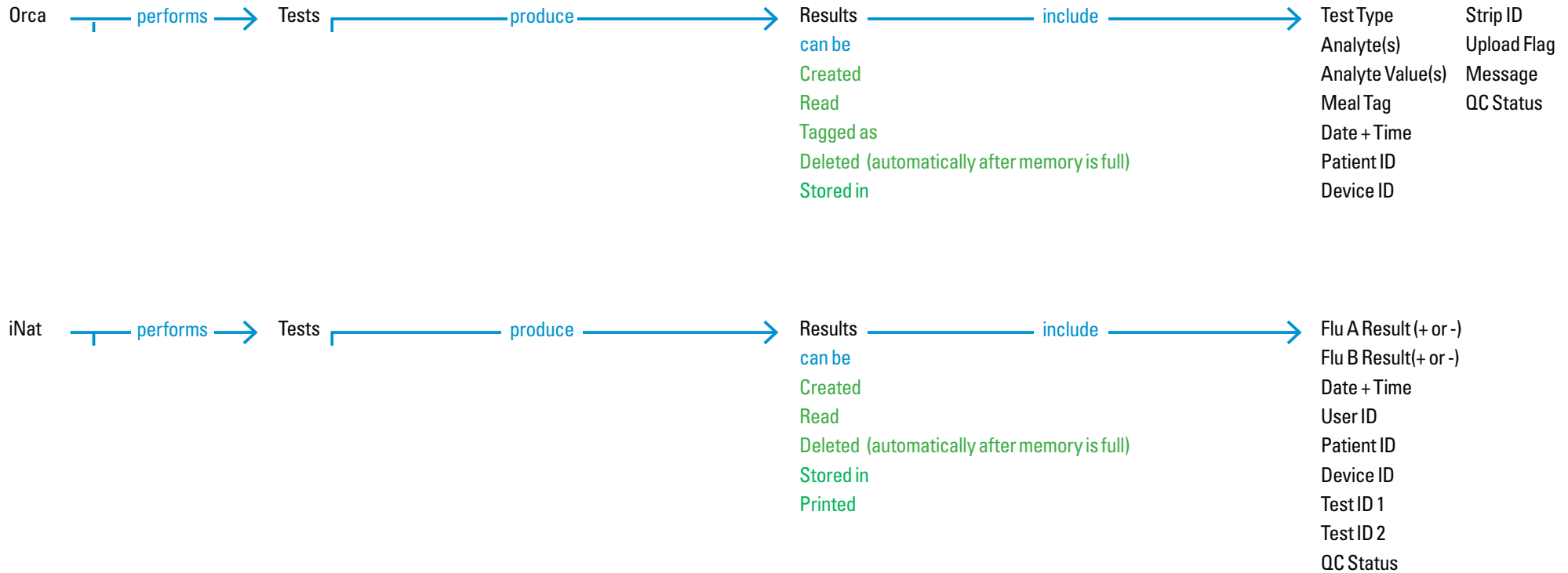
Orca: Blood Analyte Meter, revised version



iNat: Nasal Flu Meter

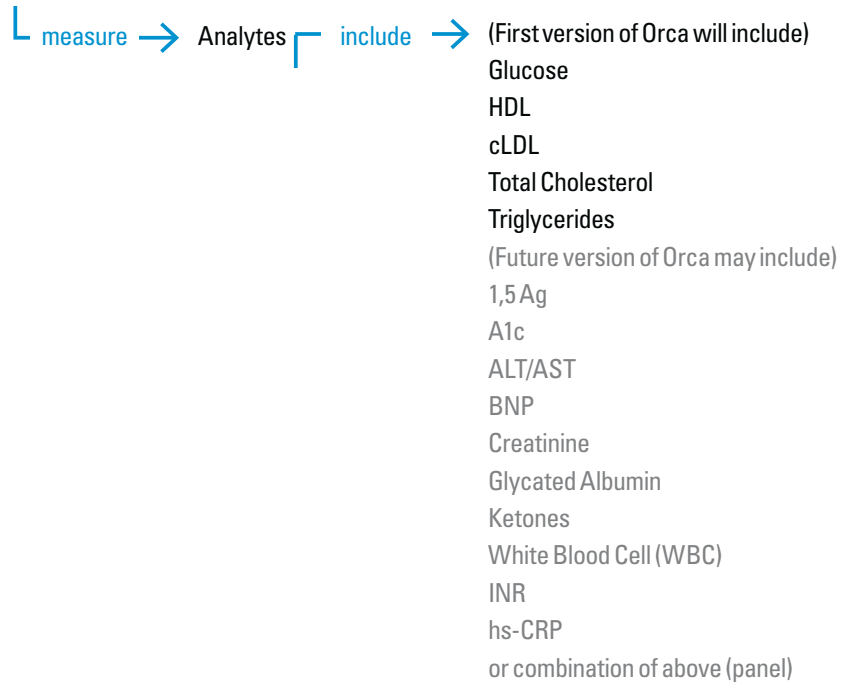


Similar primary functions



Different analytes

Orca

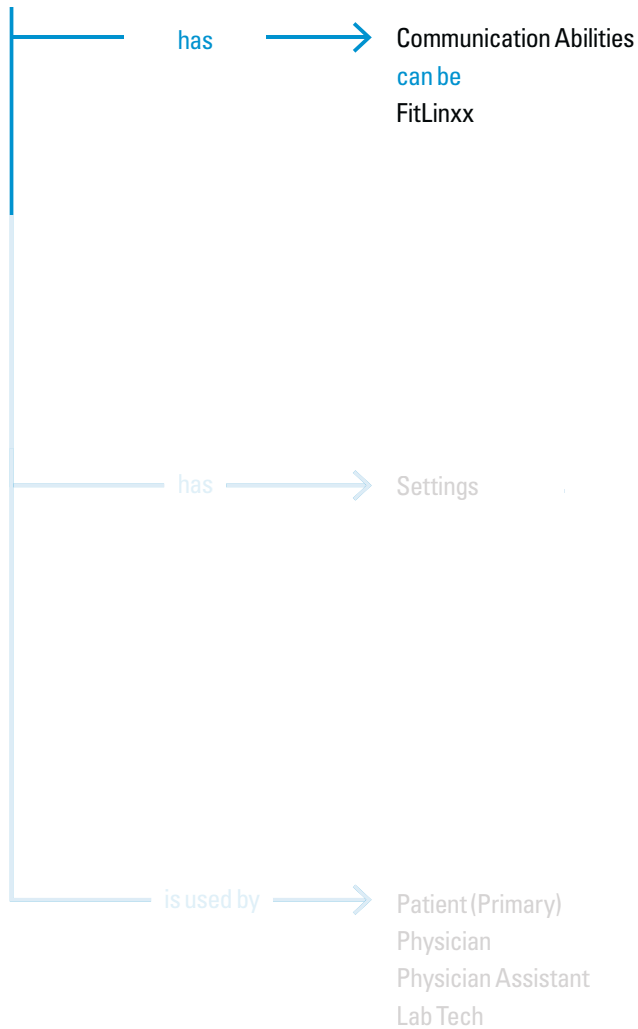


iNat

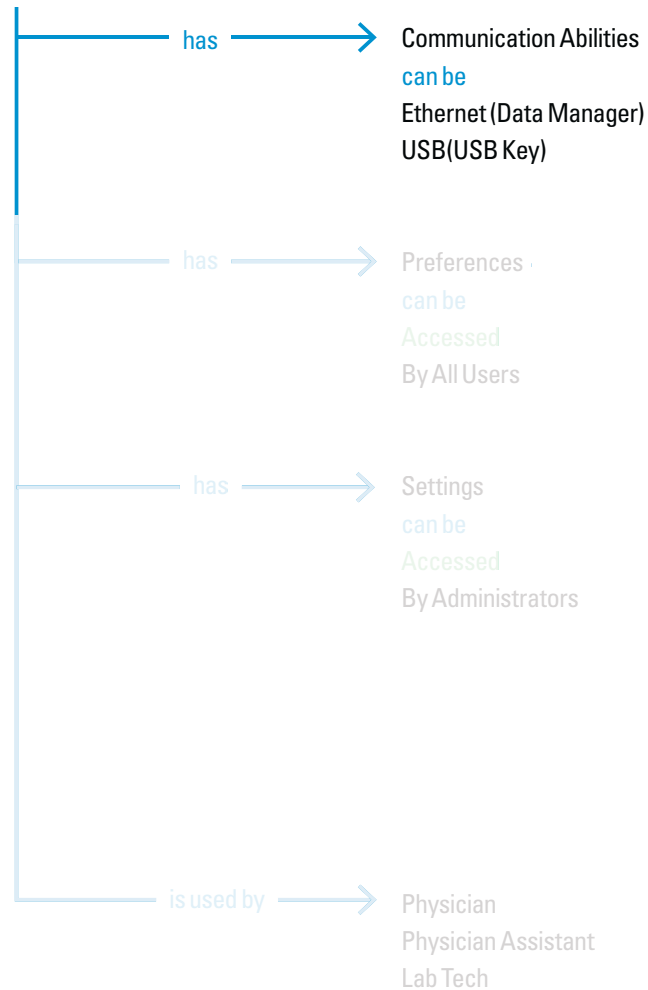


Different communication protocols

Orca

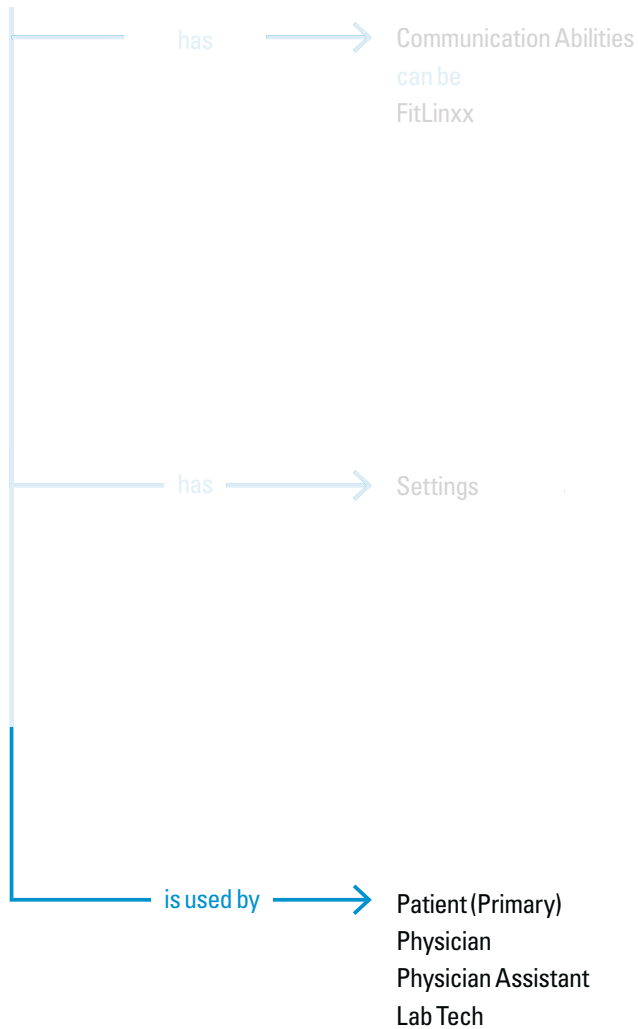


iNat

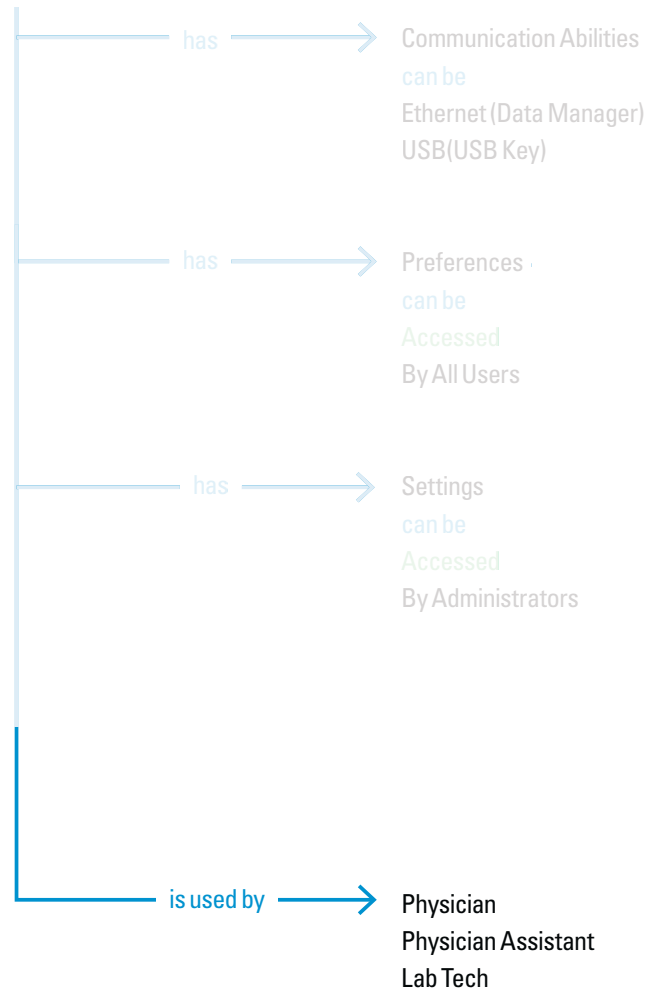


Different user types

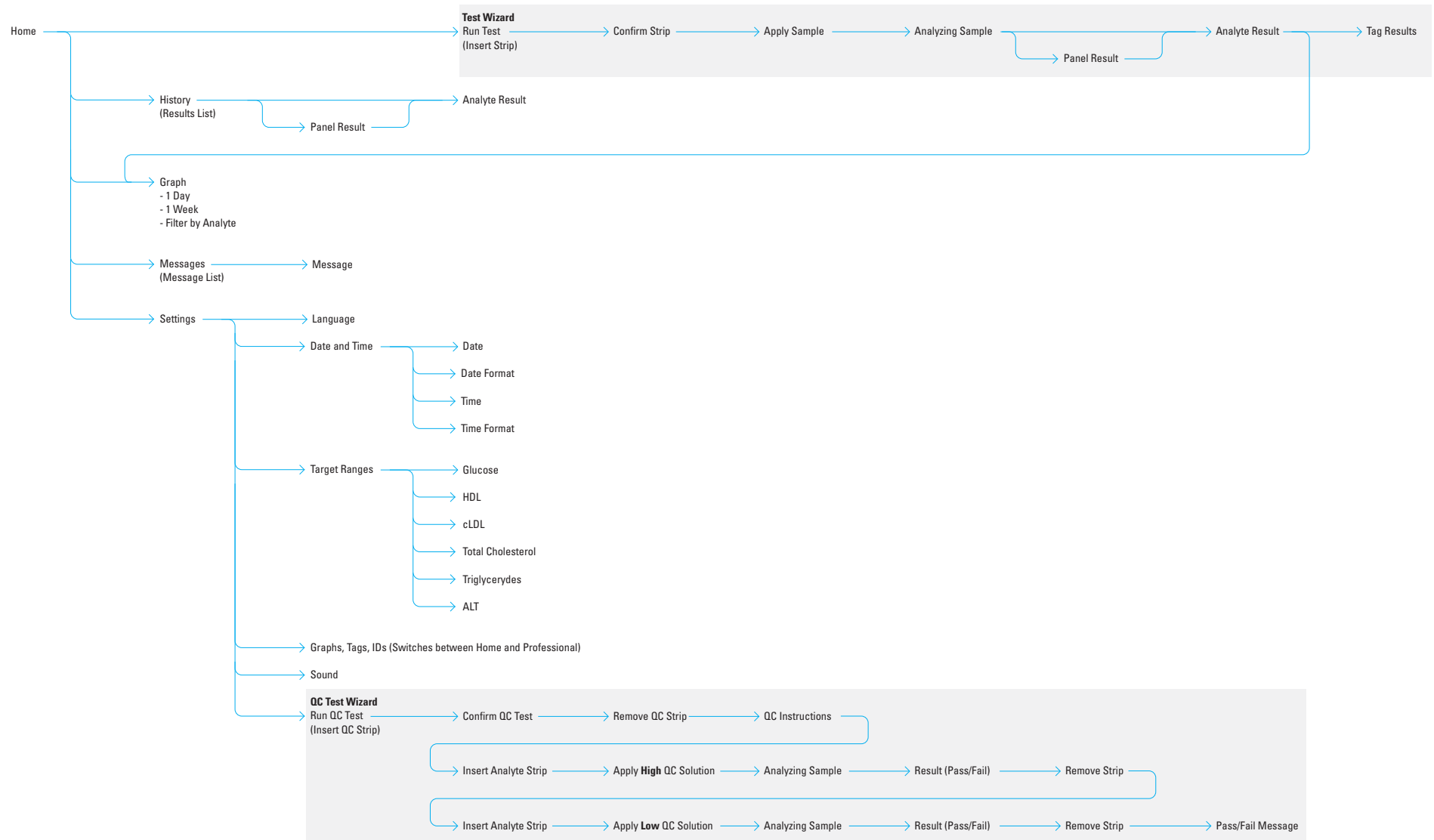
Orca



iNat

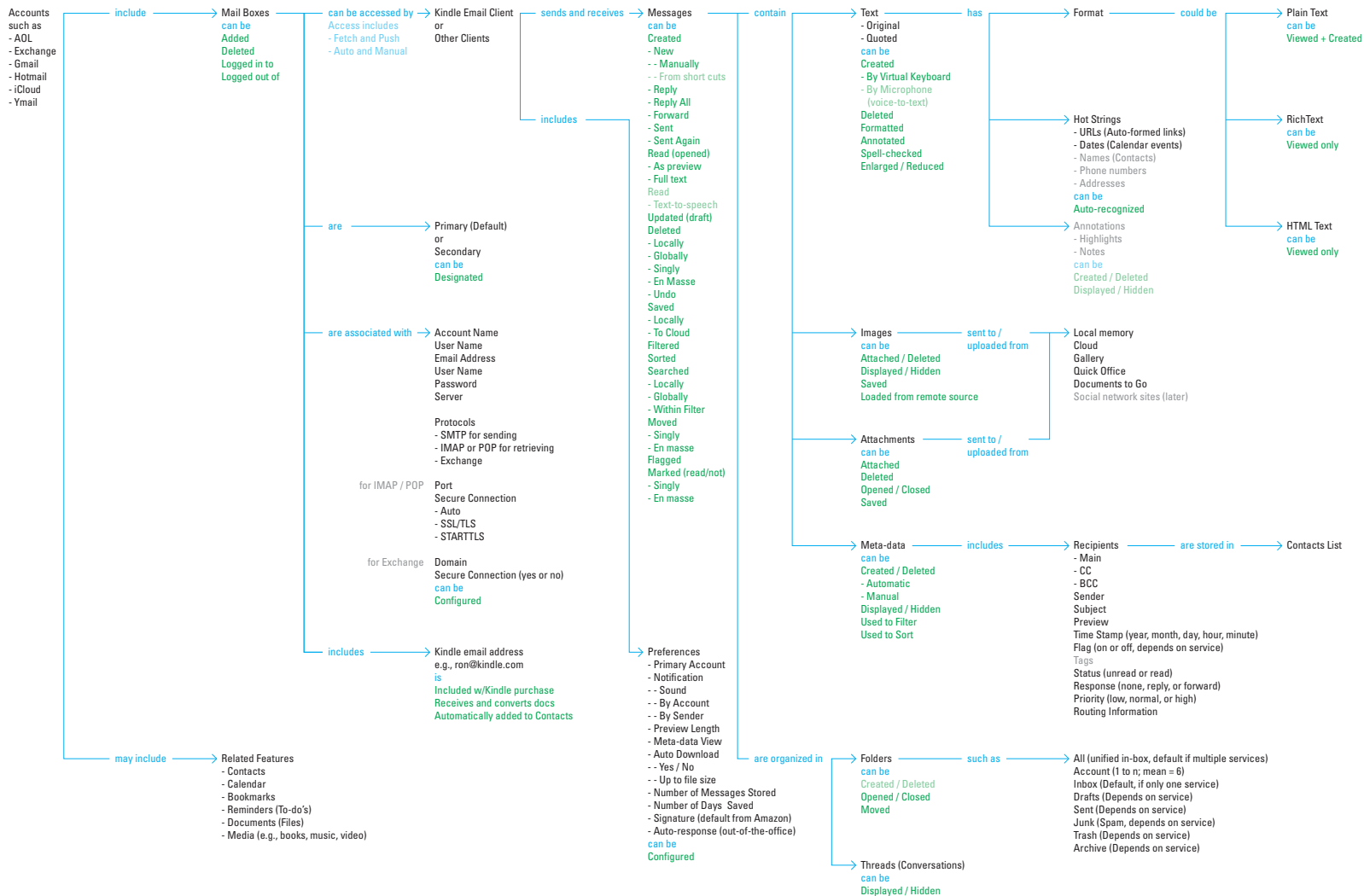


CMs (prior) are not IAs (below)



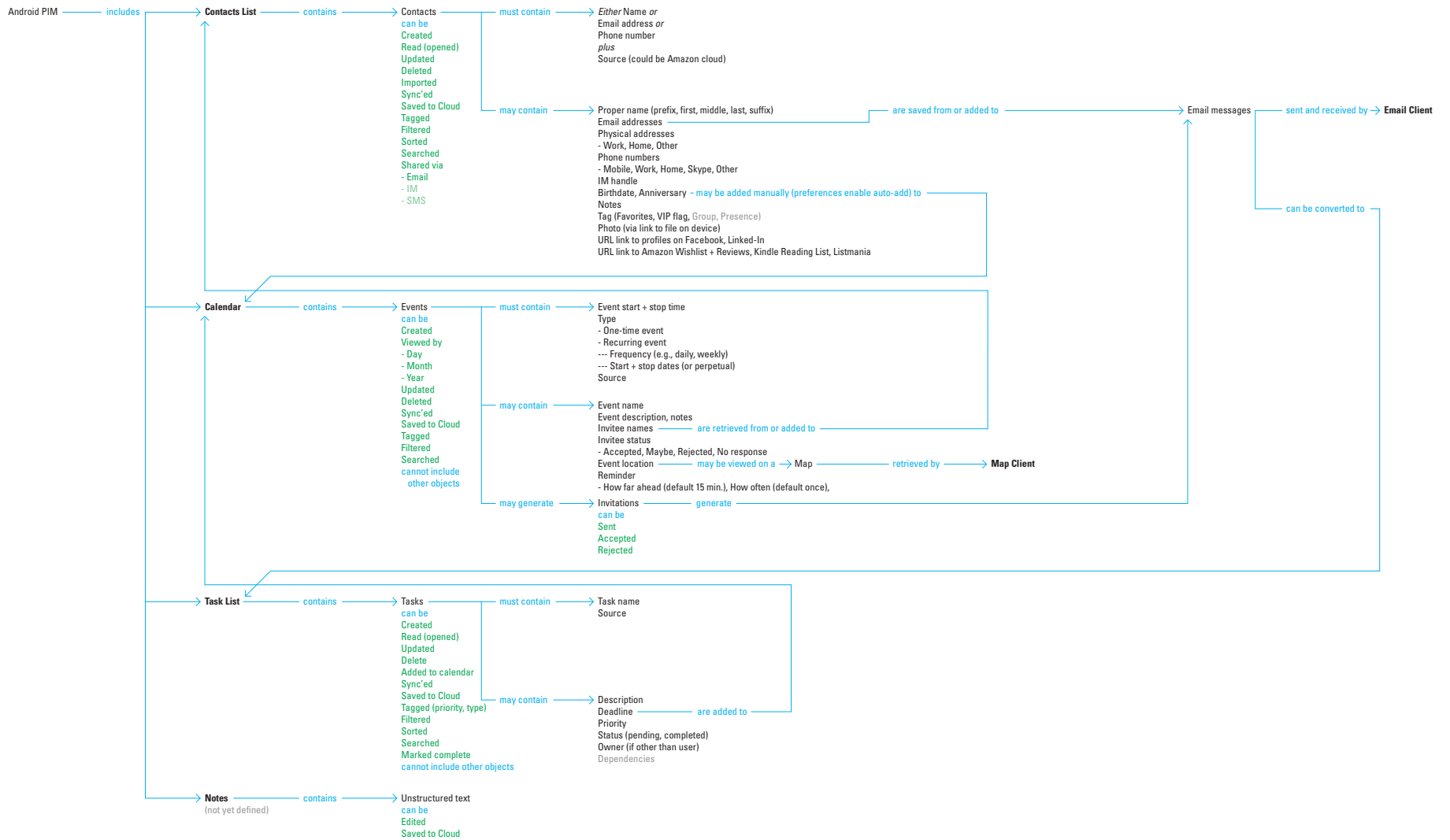
Other examples

Android, email client

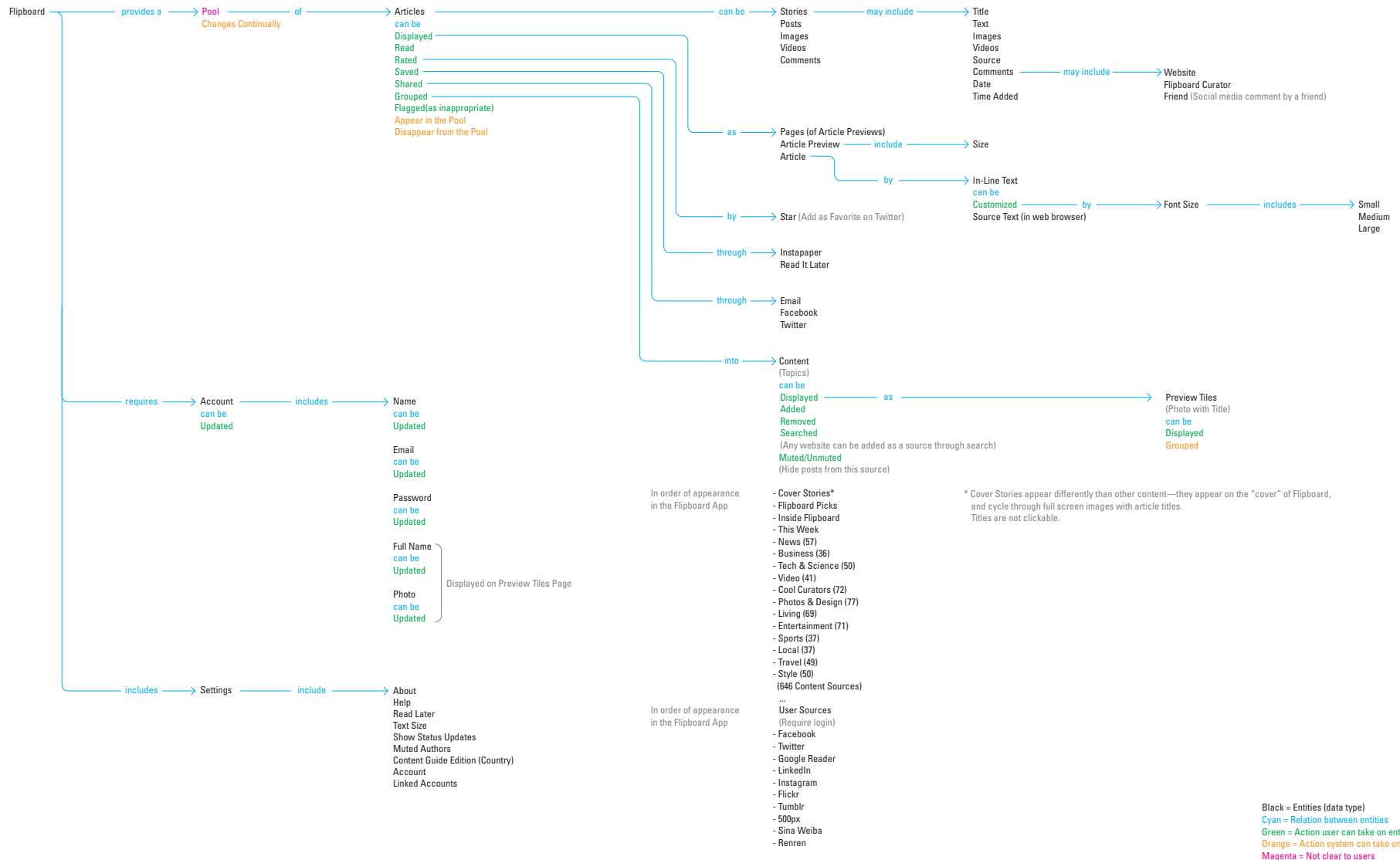


Black = Entities (data type)
 Gray = Entity to be added later
 Cyan = Relation between entities
 Green = Action user can take on entity
 Light Green = Action to be added later

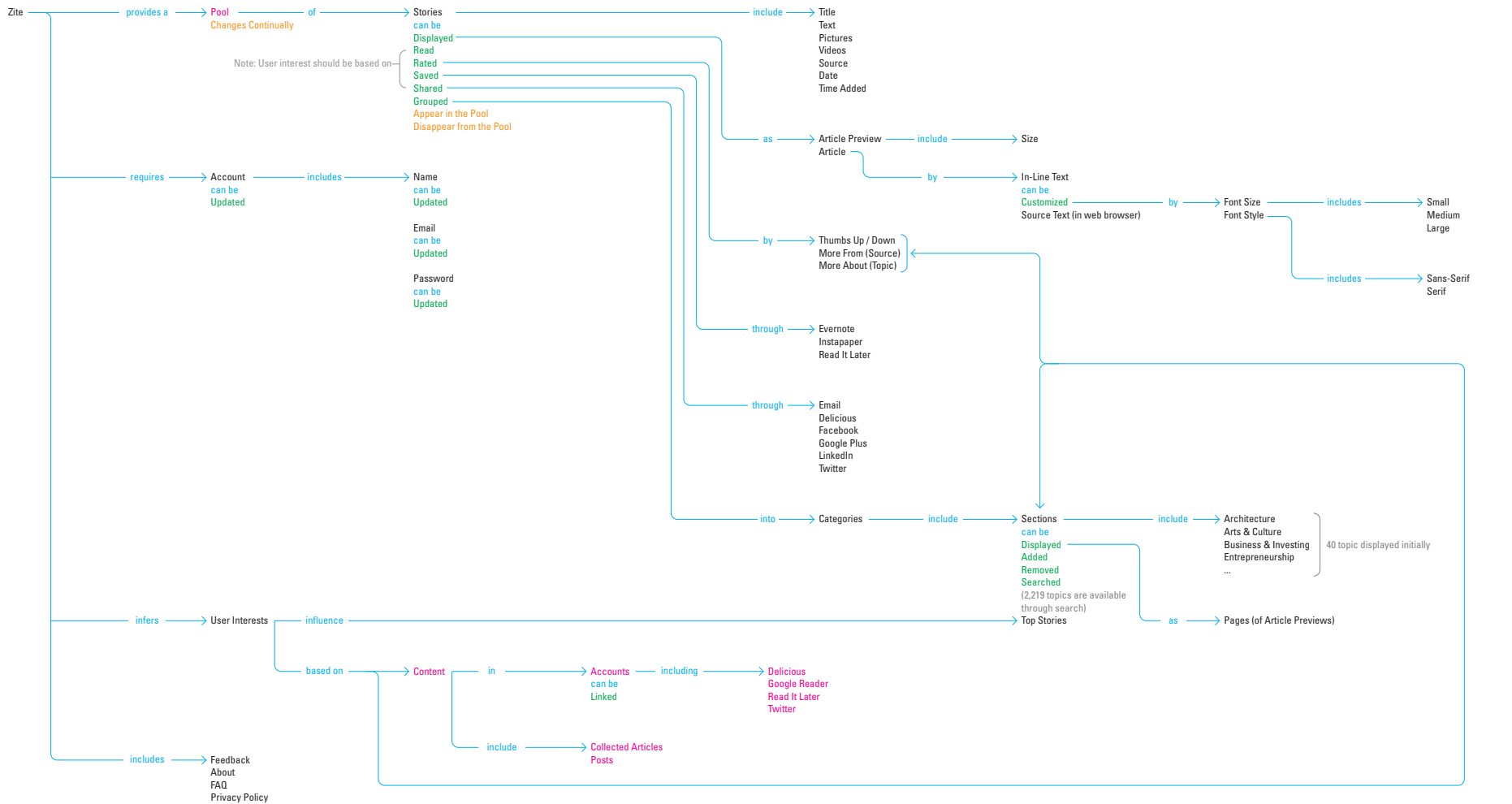
Android, PIM



Flipboard, social magazine

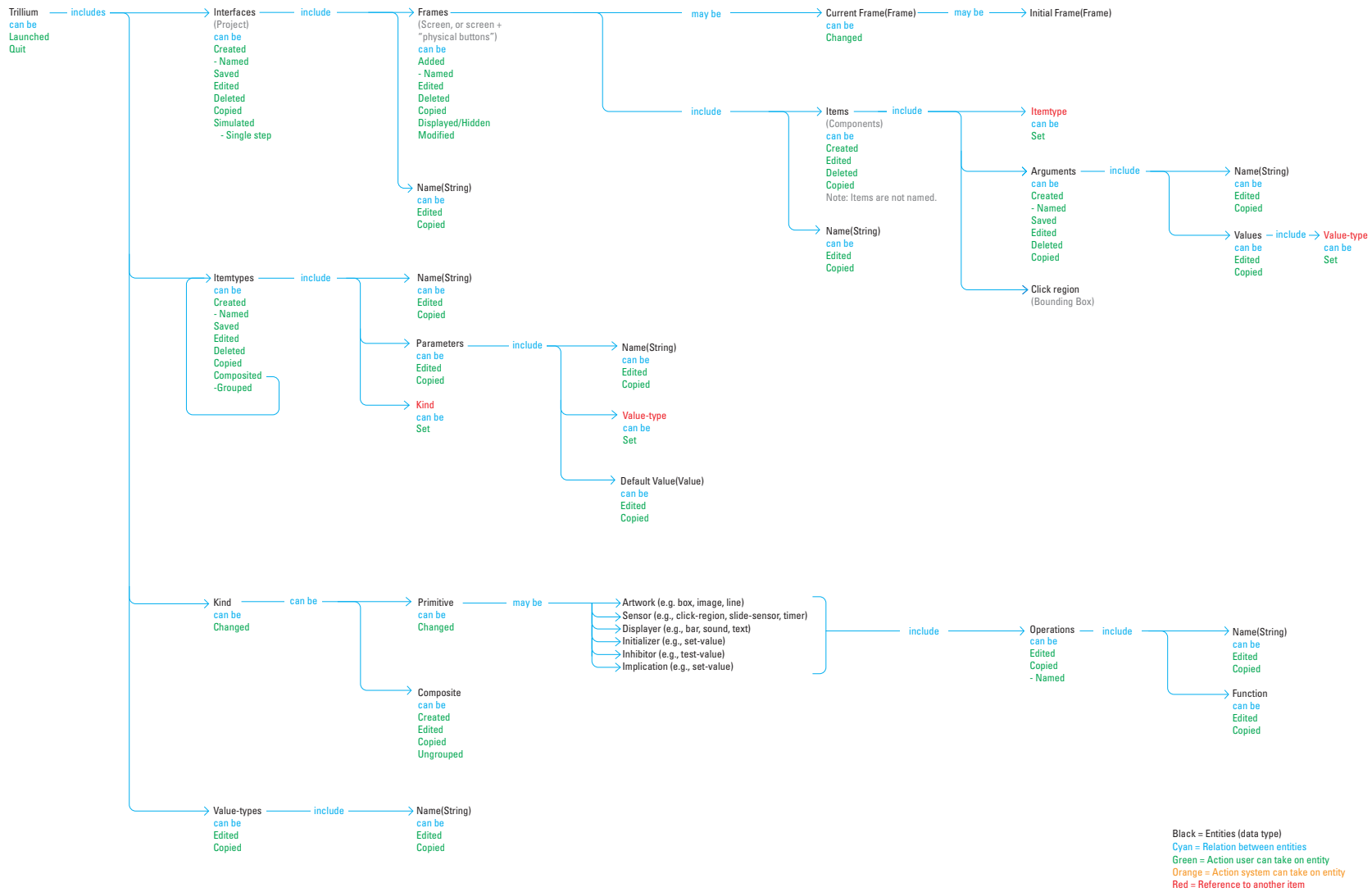


Zite, discovery engine



Black = Entities (data type)
 Cyan = Relation between entities
 Green = Action user can take on entity
 Orange = Action system can take on entity
 Magenta = Not clear to users

Trillium, Xerox copier development environment



PLX, early data model study

Dashboard

- Annotations
- Filter Widgets
- Metadata

Container

- Annotations
- Filter Widgets
- Metadata

Table foo-2

A	B	C	D	E	
Cntry	Region	Prod C	Prod D	Prod E	
1	CN	APAC	25	38	19
2	JP	APAC	37	42	33
3	KR	APAC	27	12	41
4	TW	APAC	13	18	15
5	SP	APAC	9	6	7
6	UK	EMEA	43	29	16
n	FR	EMEA	32	28	15

View	Query	Proximate	Expression Source
			SELECT columns A, C, D FROM table foo-2
		Final	Expression Source Schedule SELECT column (processes that led to A, C, D, E) FROM table foo-root BEGIN every Sunday at 23:00:00 GMT UNTIL ongoing
		Transforms	Joins Derivations SET column F = C + D + E
		Filters	WHERE country = CN or JP or KR

Result	All data																												
	Sample data Column title list																												
	<table border="1"> <thead> <tr> <th>A</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> <tr> <th>Cntry</th> <th>Prod C</th> <th>Prod D</th> <th>Prod E</th> <th>Total</th> </tr> </thead> <tbody> <tr><td>1</td><td>CN</td><td>25</td><td>38</td><td>19</td><td>82</td></tr> <tr><td>2</td><td>JP</td><td>37</td><td>42</td><td>33</td><td>112</td></tr> <tr><td>3</td><td>KR</td><td>27</td><td>12</td><td>41</td><td>80</td></tr> </tbody> </table>	A	C	D	E	F	Cntry	Prod C	Prod D	Prod E	Total	1	CN	25	38	19	82	2	JP	37	42	33	112	3	KR	27	12	41	80
A	C	D	E	F																									
Cntry	Prod C	Prod D	Prod E	Total																									
1	CN	25	38	19	82																								
2	JP	37	42	33	112																								
3	KR	27	12	41	80																								

Visualization	Type	Properties	Annotations	Filter Widgets
	Bar Chart	TBD	Text	
	Standard	Image		
				DROPDOWN = CN, JP, KR

Metadata	Title	Keywords	Description	Creator	Date Created	Permissions	Owners	Collaborators	Readers
	Quarterly Revenue		Filtered by China, Japan, Korea	Natali	August 10, 2011		Jeff	Mike	David

Quarterly Revenue

Filter by Total ▼

- China
- Japan
- Korea

Product	China	Japan	Korea	Total
Product C	19	33	30	82
Product D	33	42	37	112
Product E	41	12	27	80

Other Views

Other Containers **Other Views**

Other Dashboards **Other Containers** **Other Views**

Summary

Product teams should agree on a conceptual model— a definition of what users need to know— well before creating wire-frames and writing code.

The model will change over the course of the product development process.

Developing a standard form for conceptual models will make them easier to understand, easier to make, and easier to teach new designers.

Special thanks to
Jonathan Arnowitz
Bryan Crowe
Austin Henderson
Marci Robidoux
Mathew Varghese

hugh@dubberly.com
415 648 9799

Presentation posted at
www.dubberly.com/presentations/conceptual_models.pdf