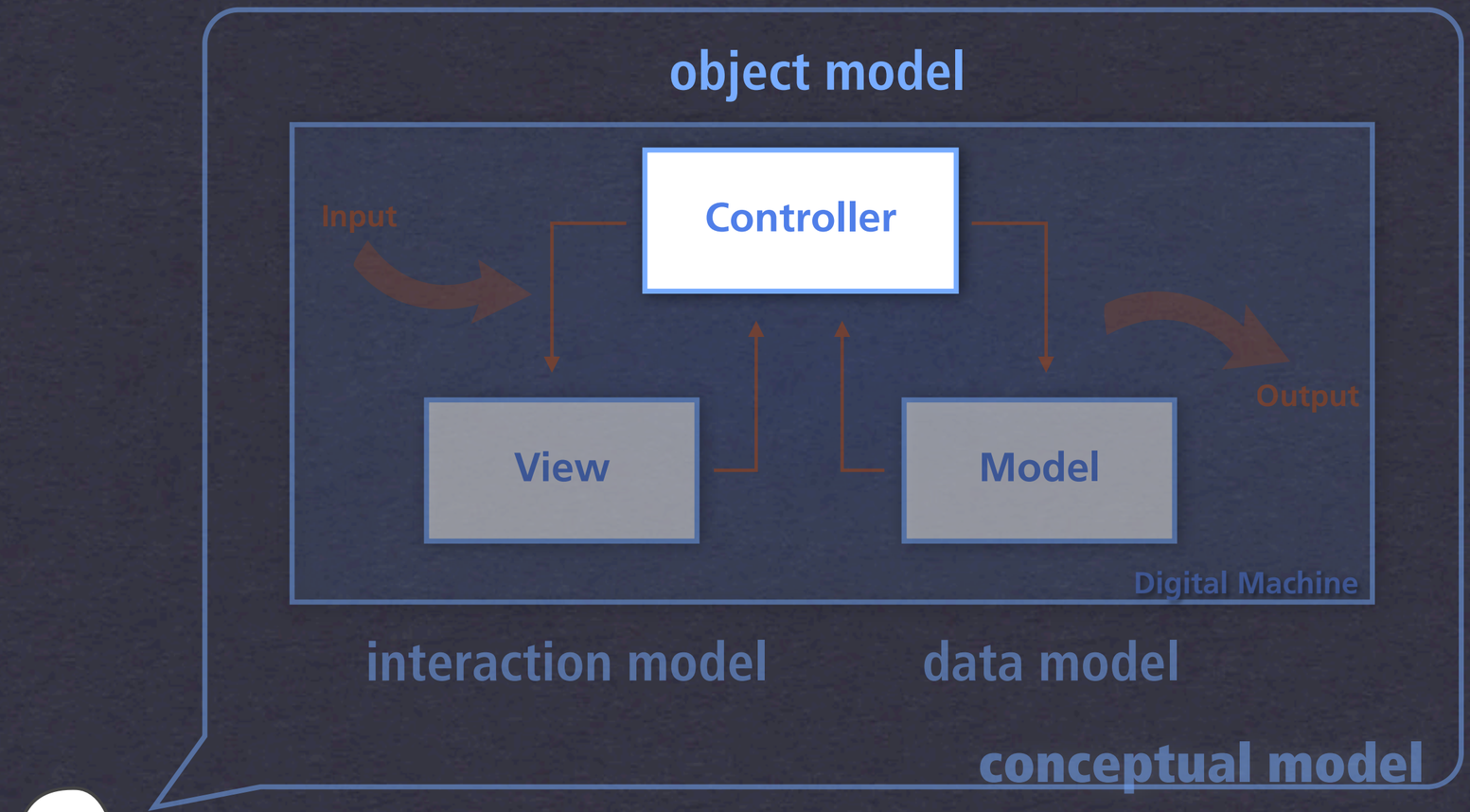
Object Model

How does this thing work?

№ 5, *Design of Digital Machines*
Tim Sheiner

The four sub models of the machine



The object model corresponds most closely to the controller.

How does the thing work?

The object model defines system logic

- what are the functional objects and what do they do?
- how are these objects connected to each other?
- what is the currency they exchange?

The object model is narrative

The object model is like a story so you can use narrative analytic techniques to design or uncover it.

For example:

1. Tell a story about system with the same purpose as the machine you are designing
2. Identify the characters in the story, what they care about, who they speak to and what they can do
3. The complicated character is the human part of your system, the flat, one-dimensional characters are the software objects

Some common object classes

- **Document**

  the thing the user creates with the system
- **Container**

  groups or organizes a list of other objects
- **Navigator**

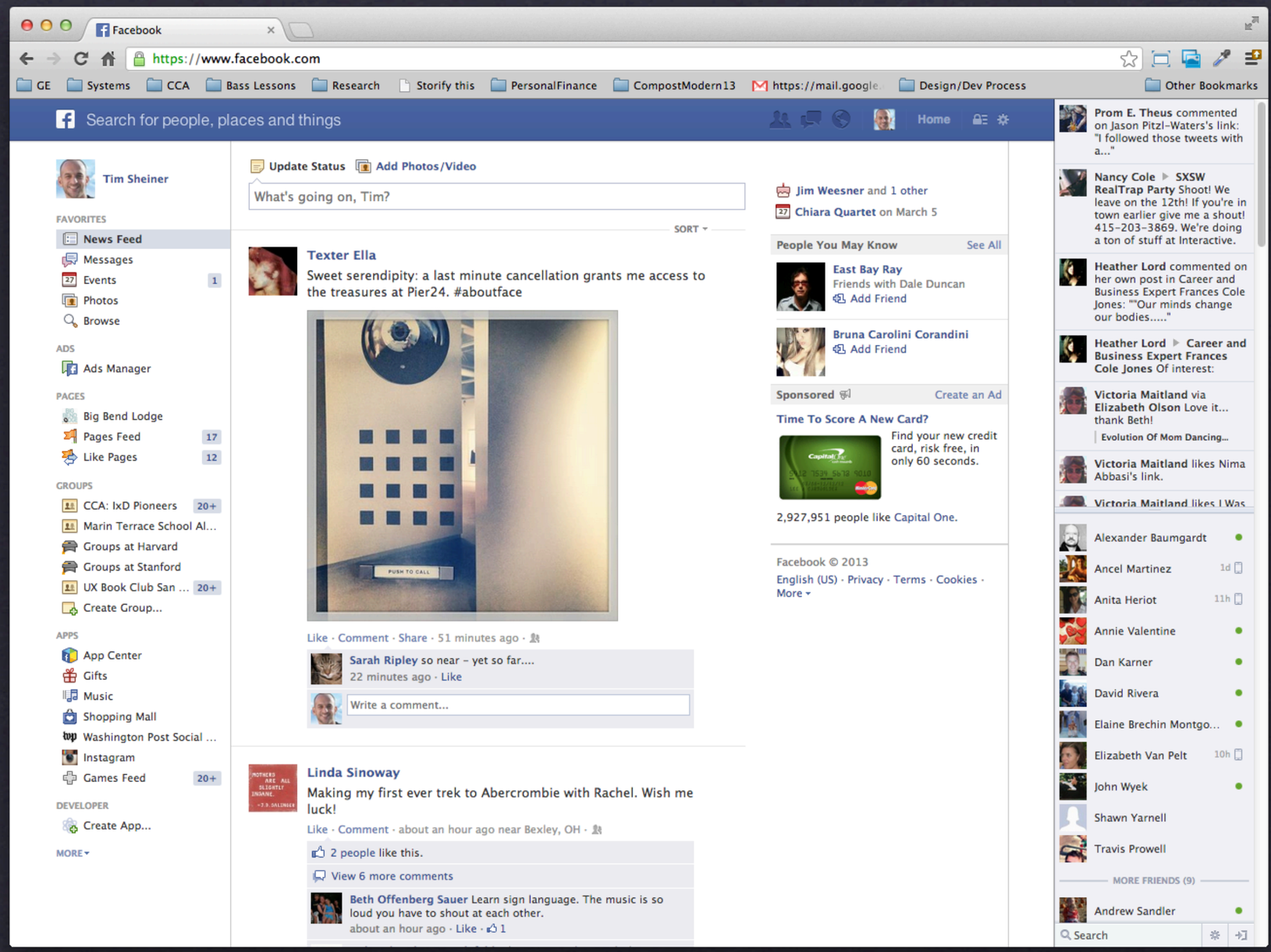
  enables movement between views, workflows or objects
- **Editor**

  enables a transformation on an object
- **Reporter**

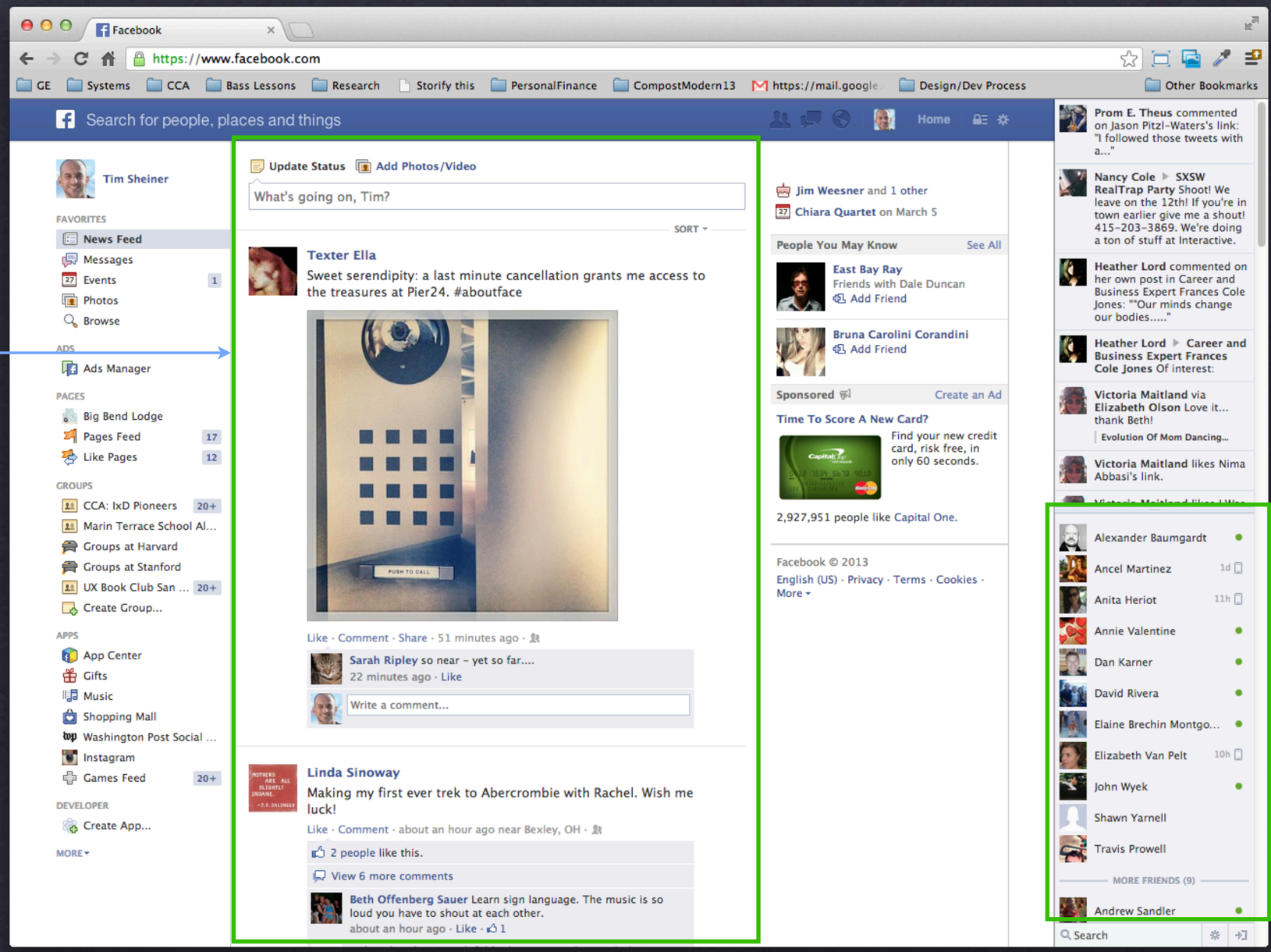
  presents read only information about an object
- **Connector**

  translates object metadata from one format to another

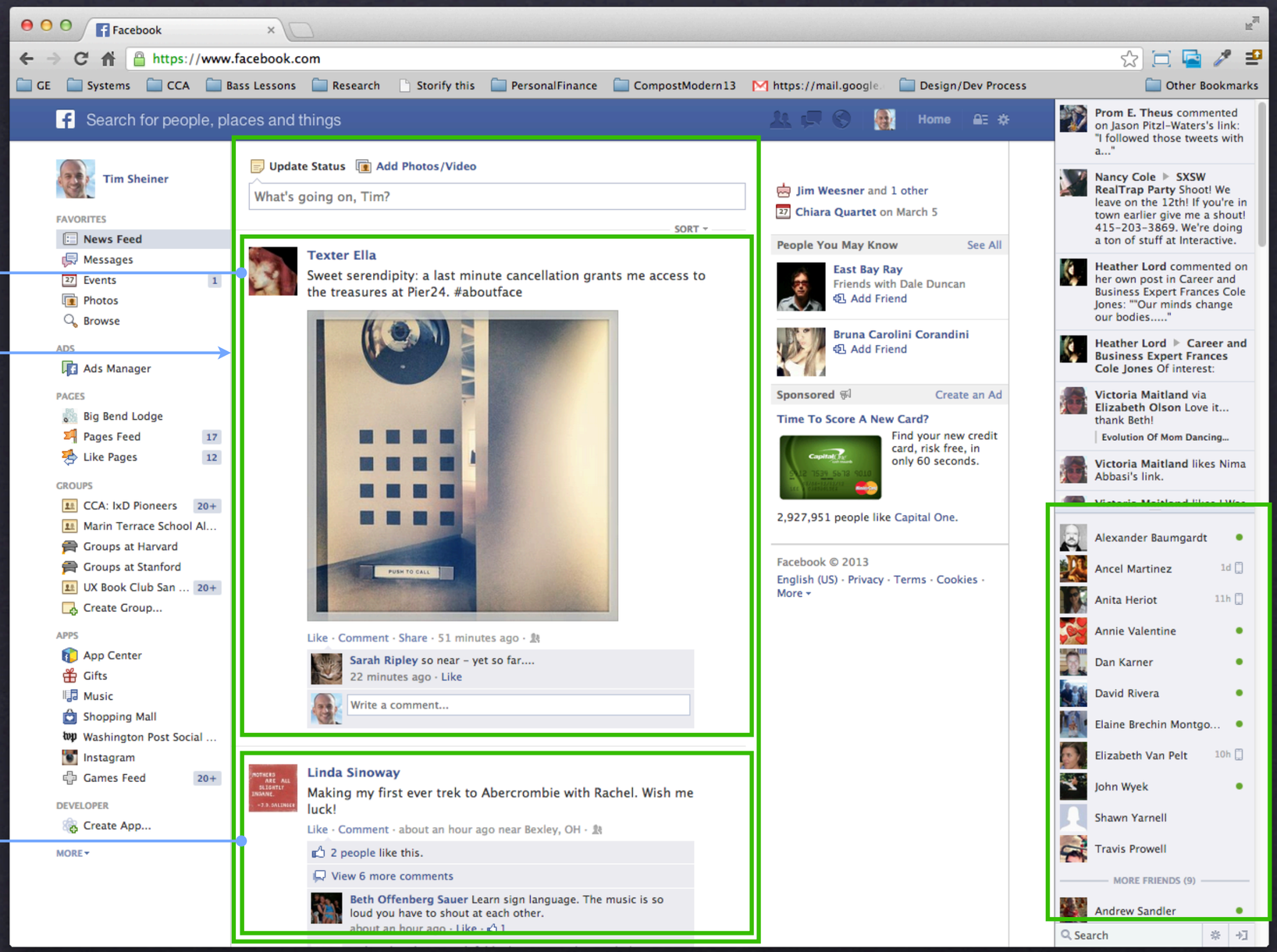
Find the objects: social network



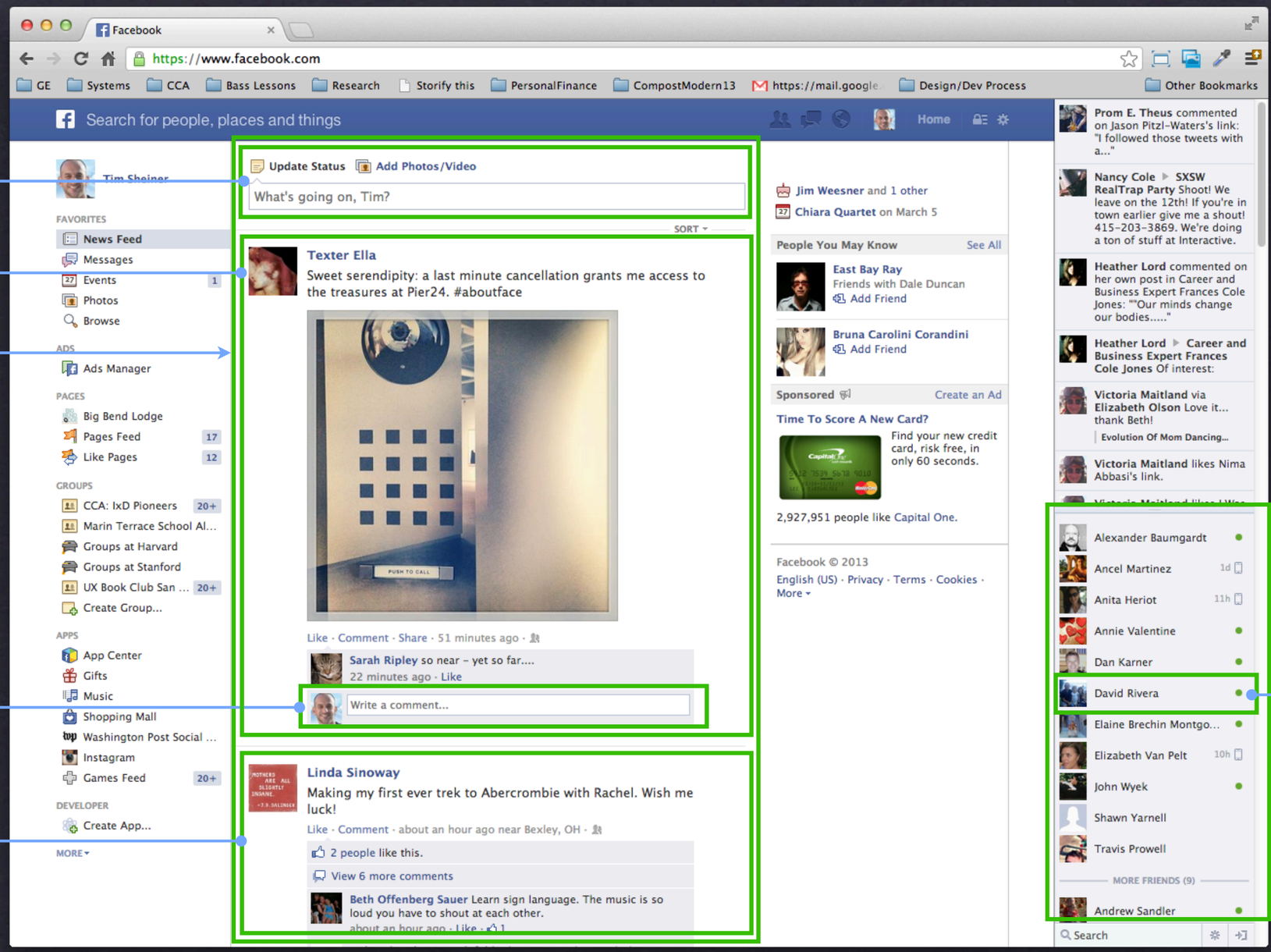
Find the objects: social network



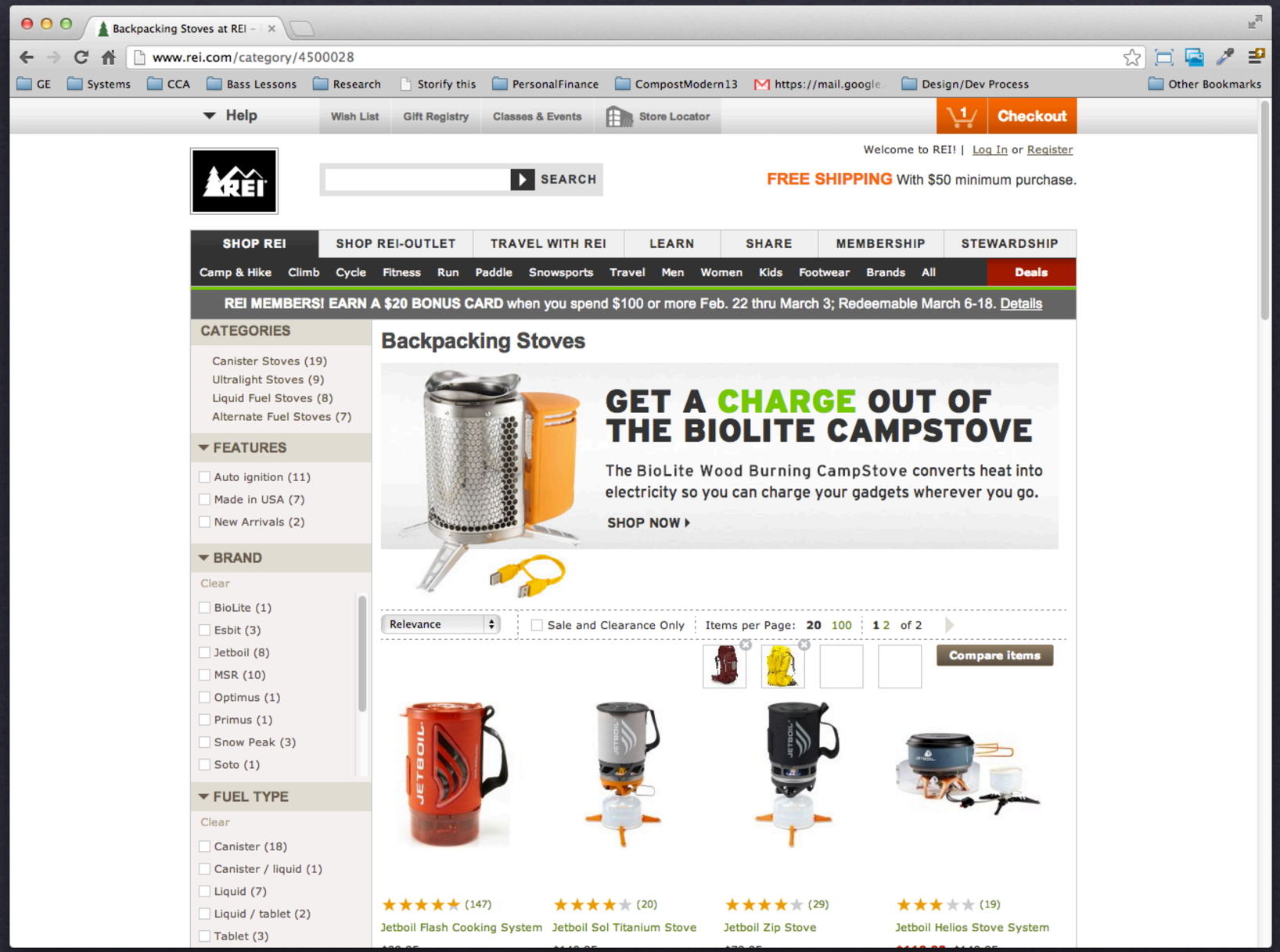
Find the objects: social network



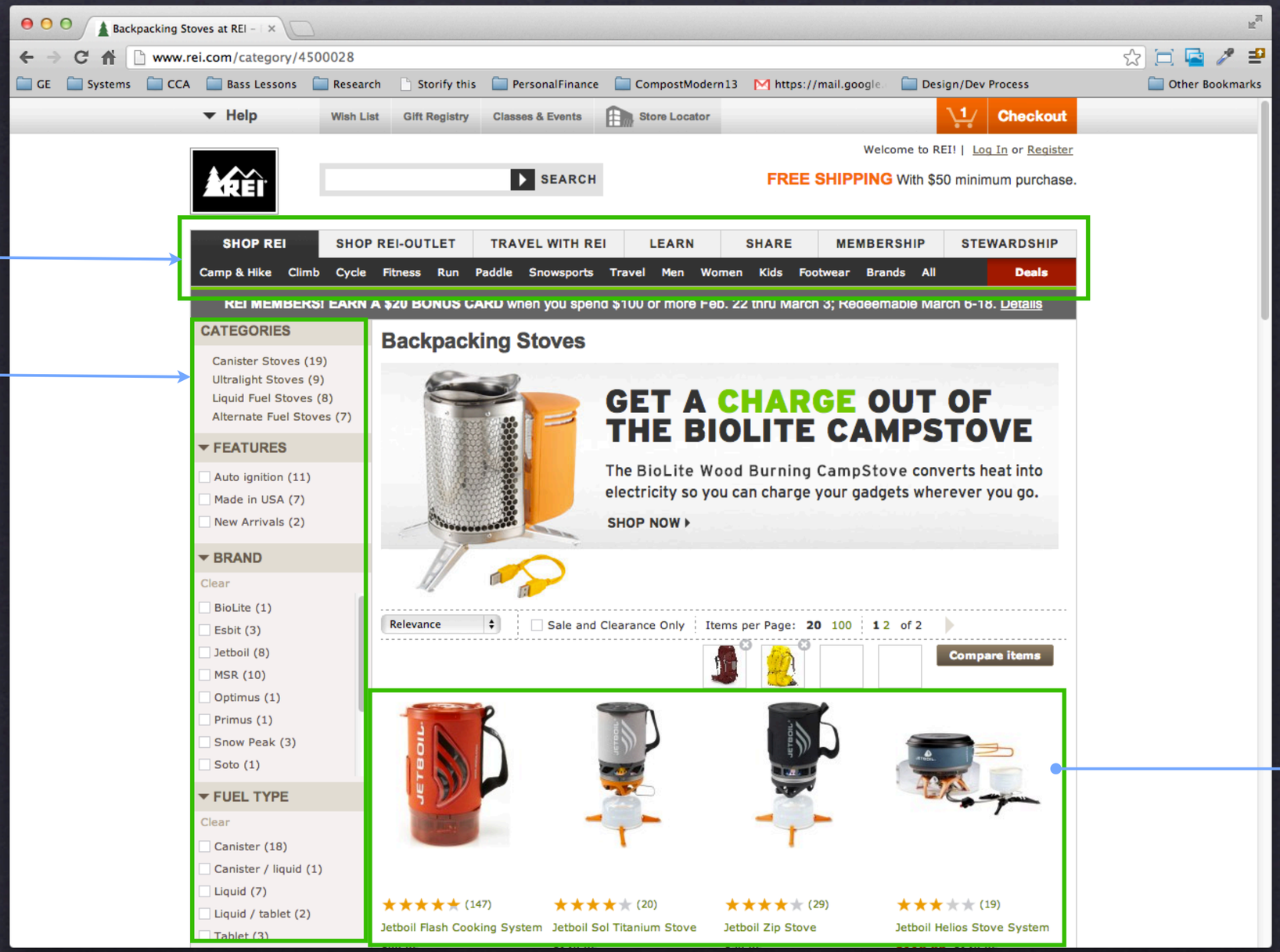
Find the objects: social network



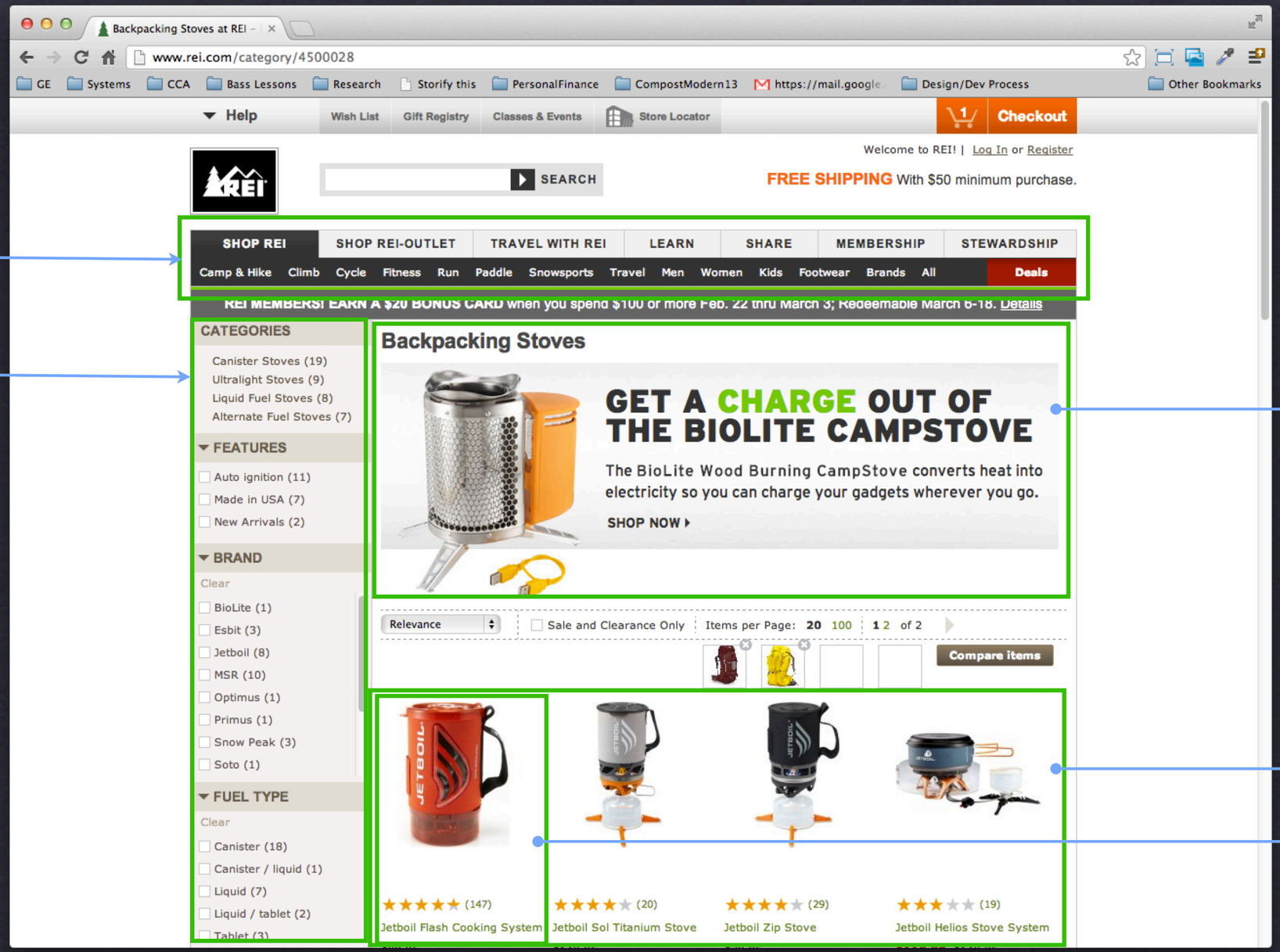
Find the objects: e-commerce



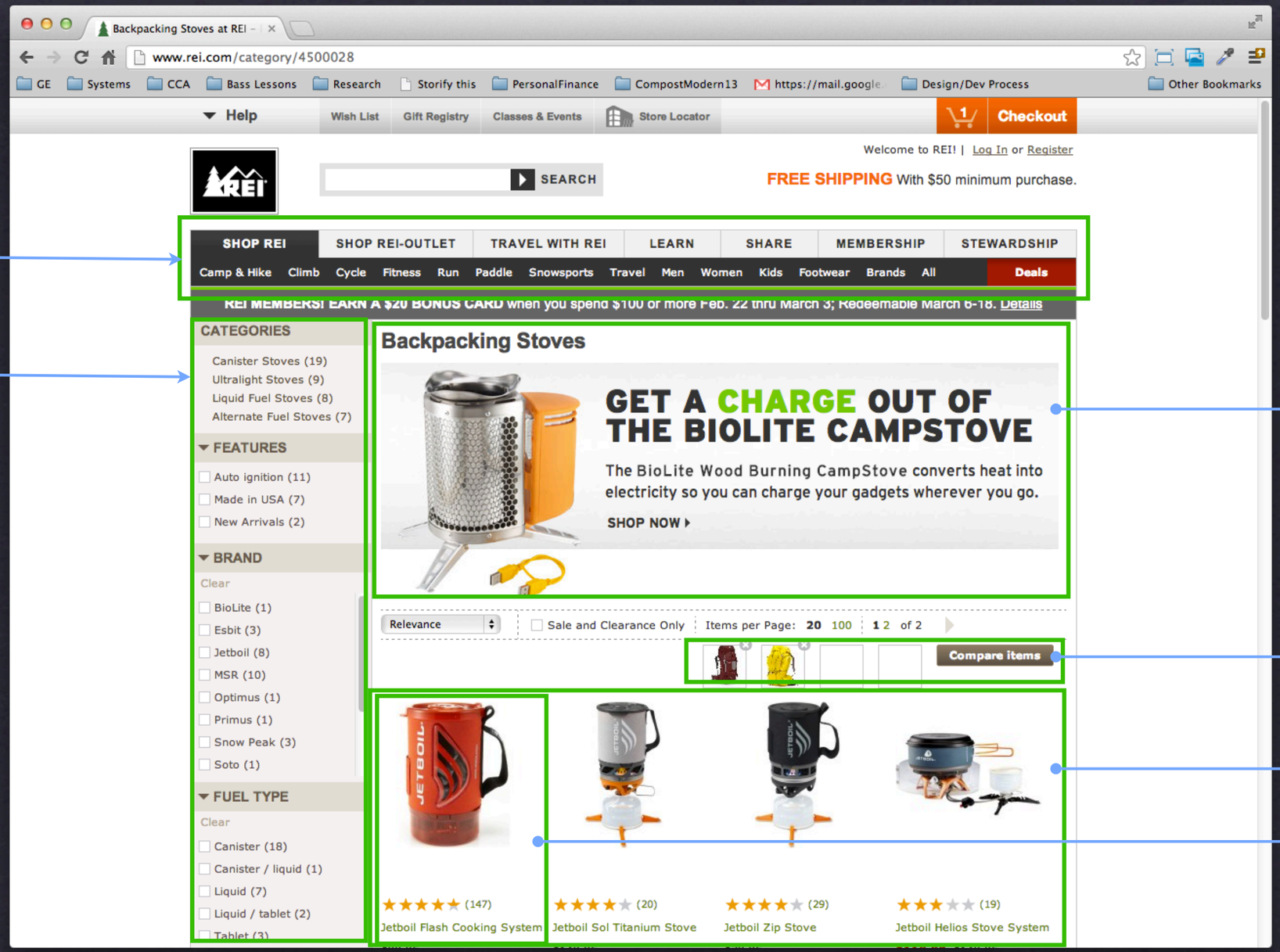
Find the objects: e-commerce



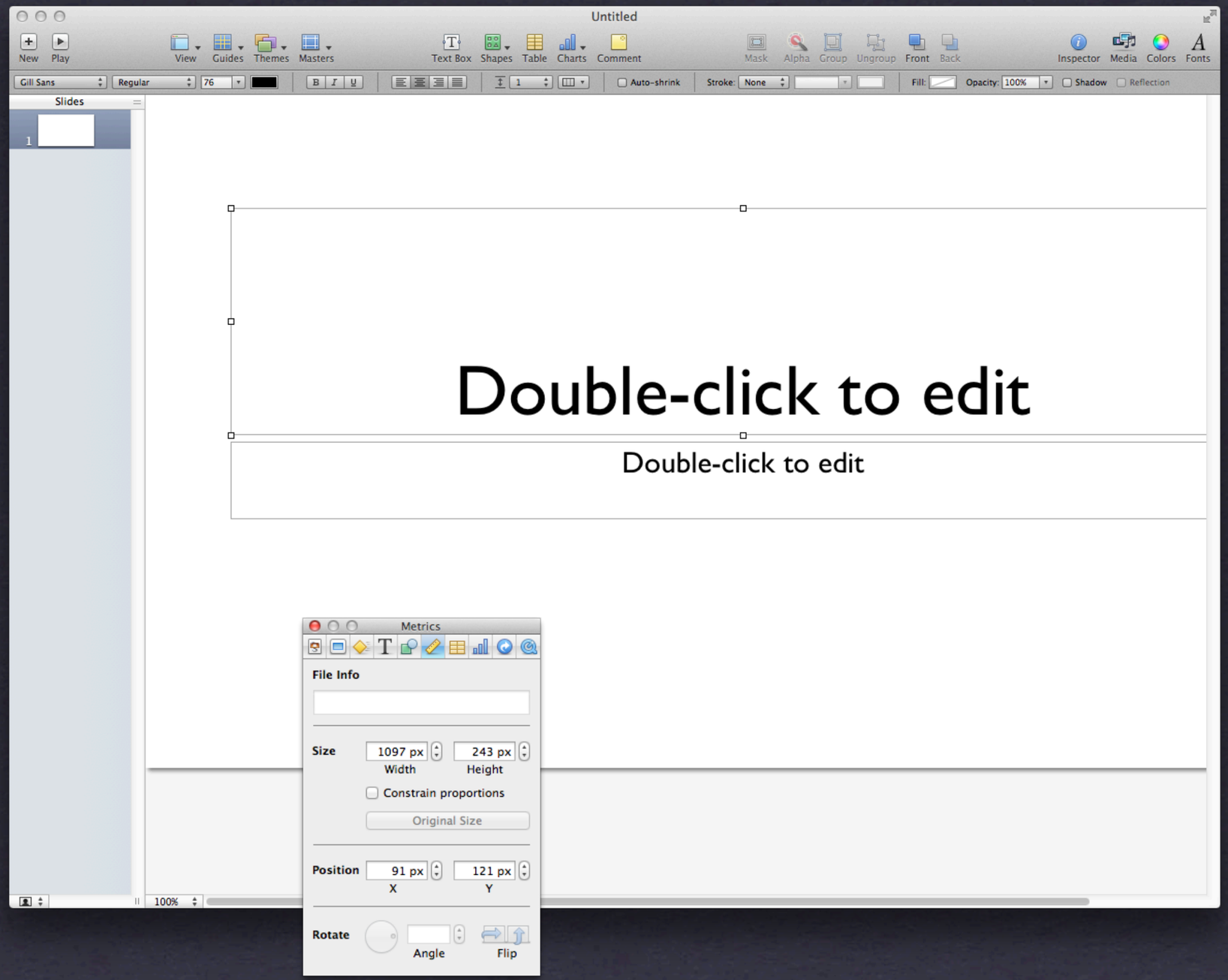
Find the objects: e-commerce



Find the objects: e-commerce



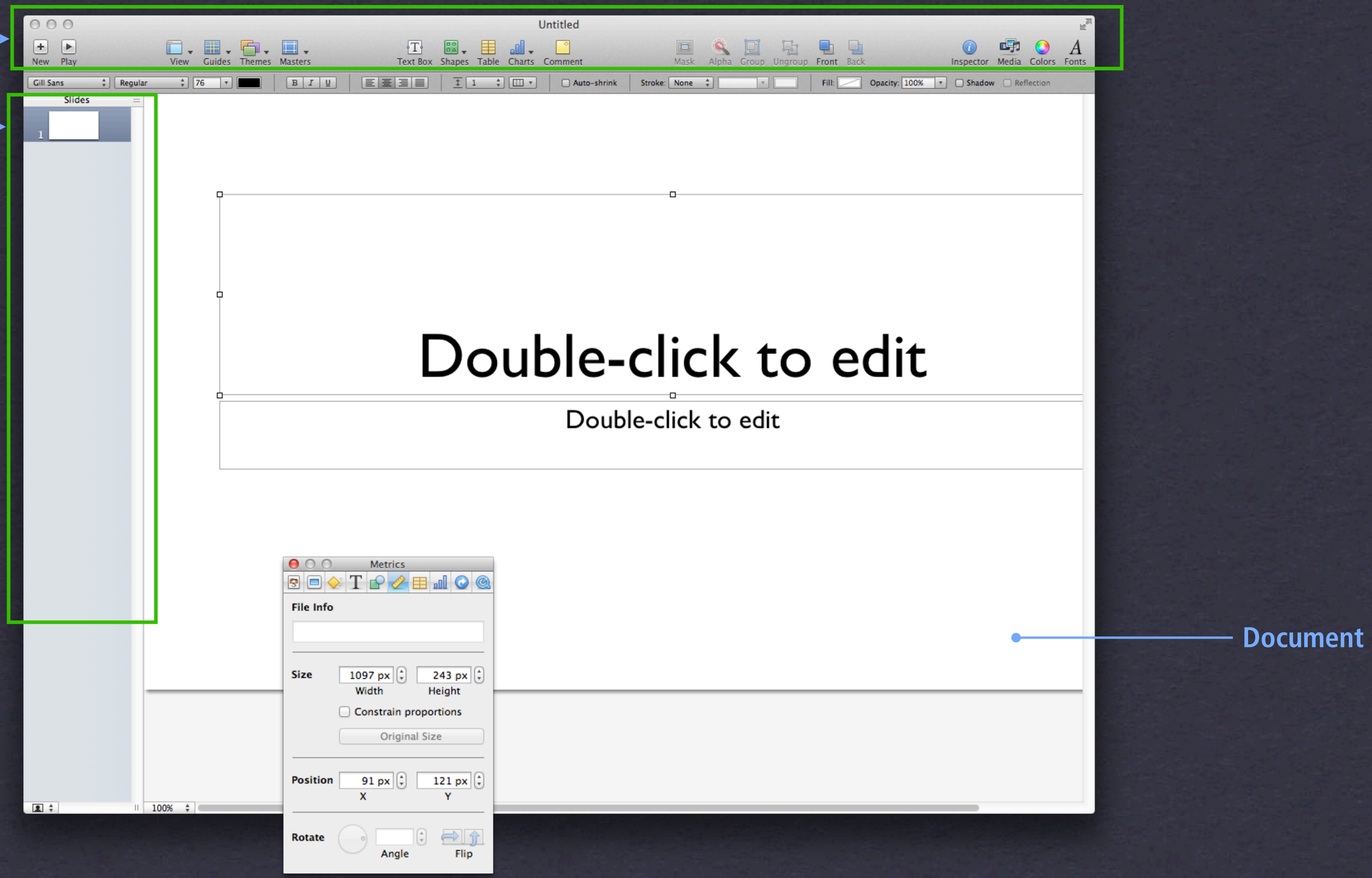
Find the objects: designer app



Find the objects: designer app

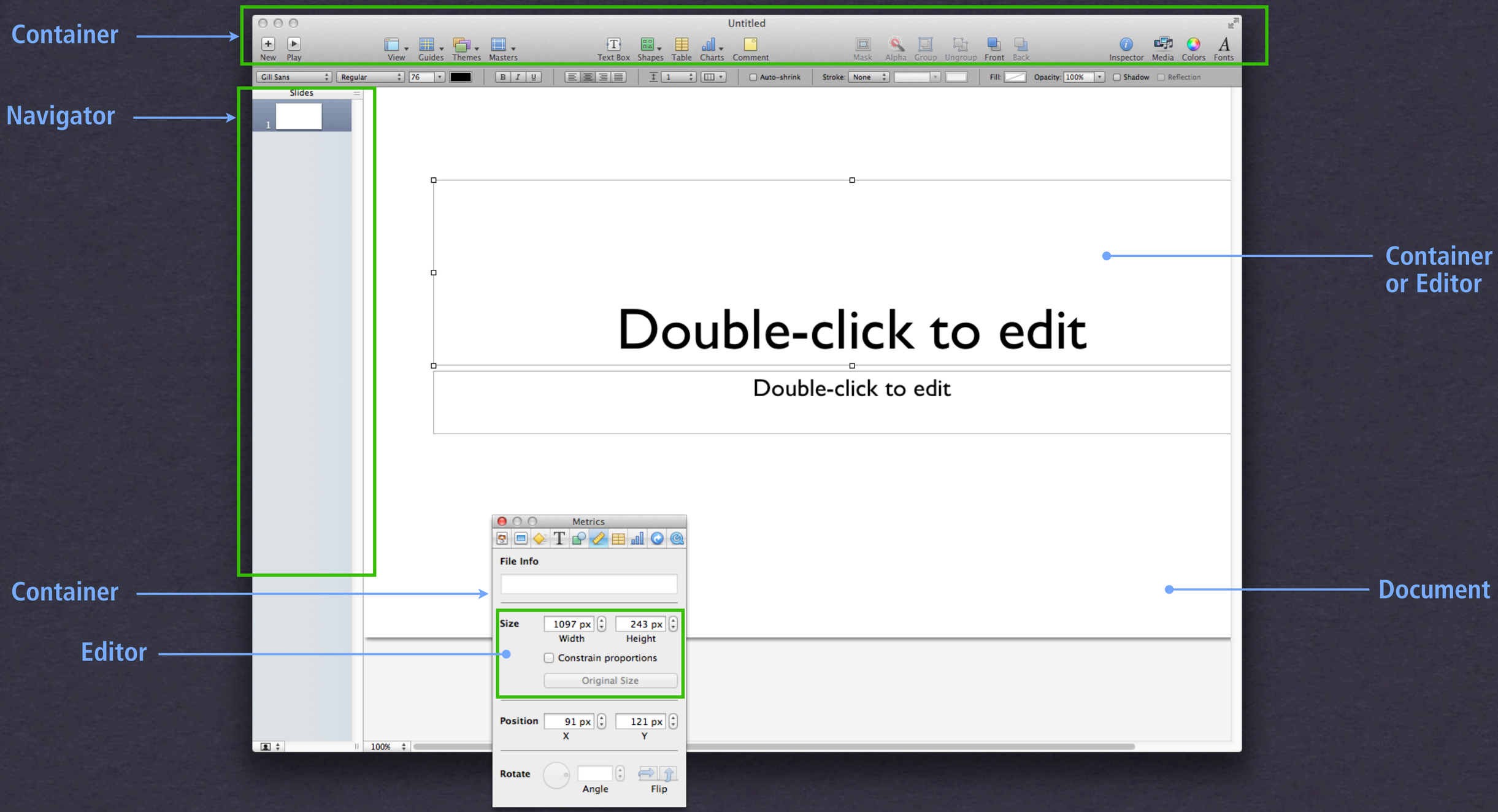
Container

Navigator



Document

Find the objects: designer app



The object definition

- **Name**

  Usually descriptive of the object's role in the system

- **Description**

  Elaborates the value or utility of the object; why it is required

- **Attributes**

  The object's metadata, the values that make one instance unique compared to another instance of the same object

- **Operations**

  Usually what the user can do to the object, sometimes what the object can do to other objects

- **Privacy**

  What roles can access what operations of this object

extension of Johnson & Henderson, Conceptual Models, 2012

Communicating the Object Model

Several standard forms

- prose list
- table
- illustration

Object Model Example

messageMe Objects

messageMe's objects are the source of all application functionality. The objects themselves reside within the messageMe instance, with each conversation having instances of of the objects available to it. In the initial conception, all conversations have access to all objects, however, restrictions could be imposed to create differentially valued offerings.

Activity Viewer

The activity viewer object contains and displays conversation events. Its rendered appearance will be dependent on the view and filter assigned to it. Examples of possible views include a timeline that lists events sequentially by time, and a calendar that arranges events in the familiar calendar grid view. Additionally, the activity object can be filtered in terms of any of the parameters assigned to events. An activity widget defines a specialized combination of a view and filter for a particular use, for example, a To Do List.

Notebook

The notebook object permits individual to create private events. In general its functionality will be very similar to the Activity object but it will likely support a reduced set of events.

Messaging

The messaging object permits text content to be exchanged between objects and members. There will be various types of messages, including chat, email, posts and notifications, each with particular parameters and views. The messaging object may have the unique ability to be embedded within the activity object so that, for example, chats may be viewed within a timeline context.

Collaboration

The collaboration object is designed to support realtime co-working. As this time it is believed that it will be based largely upon the Etherpad open source project, potentially with an additional layer for adding images and post-it note objects.

File Viewer

The file viewer object is for displaying documents for conversation discussion. It must be able to import and display a large variety of standard formats from a variety of sources including the user's local machine to cloud services.

Screen Viewer

The screen viewer is for real time screen sharing.

Meetings

The meetings object is for scheduling and managing voice and video calls. It should meet existing meeting object standards by including elements like invitations, various scheduling options (e.g. one-time, all day, repeating), reminders and have the ability to call participants.

Tasks

The tasks object permits members to assign tasks to themselves and other members. Tasks will have a set of relevant parameters like subject, description, label and status.

Finder

The finder object is a special case of the messaging object that when used allows one user to send an urgent notification to another user. The notification content cannot be edited, and simply directs the receiving user to contact the sending user as soon as possible.

Prose List

Familiar information presentation for all users, but must actually be read to have value.

Does not include a visual presentation of object connections so generally needs to be supplemented with some kind of illustration.

Object Model Example*

Table 4.1: Object/operations analysis for simple office calendar application.

Objects	Attributes	Operations
Calendar	owner, current focus	examine, print, create, add event, delete event
Event	name, description, date, time, duration, location, repeat, type (e.g. meeting)	examine, print, edit (attributes)
To-Do item	name, description, deadline, priority, status	view, print, edit (attributes)
Person	name, job-description, office, phone	send email, view details

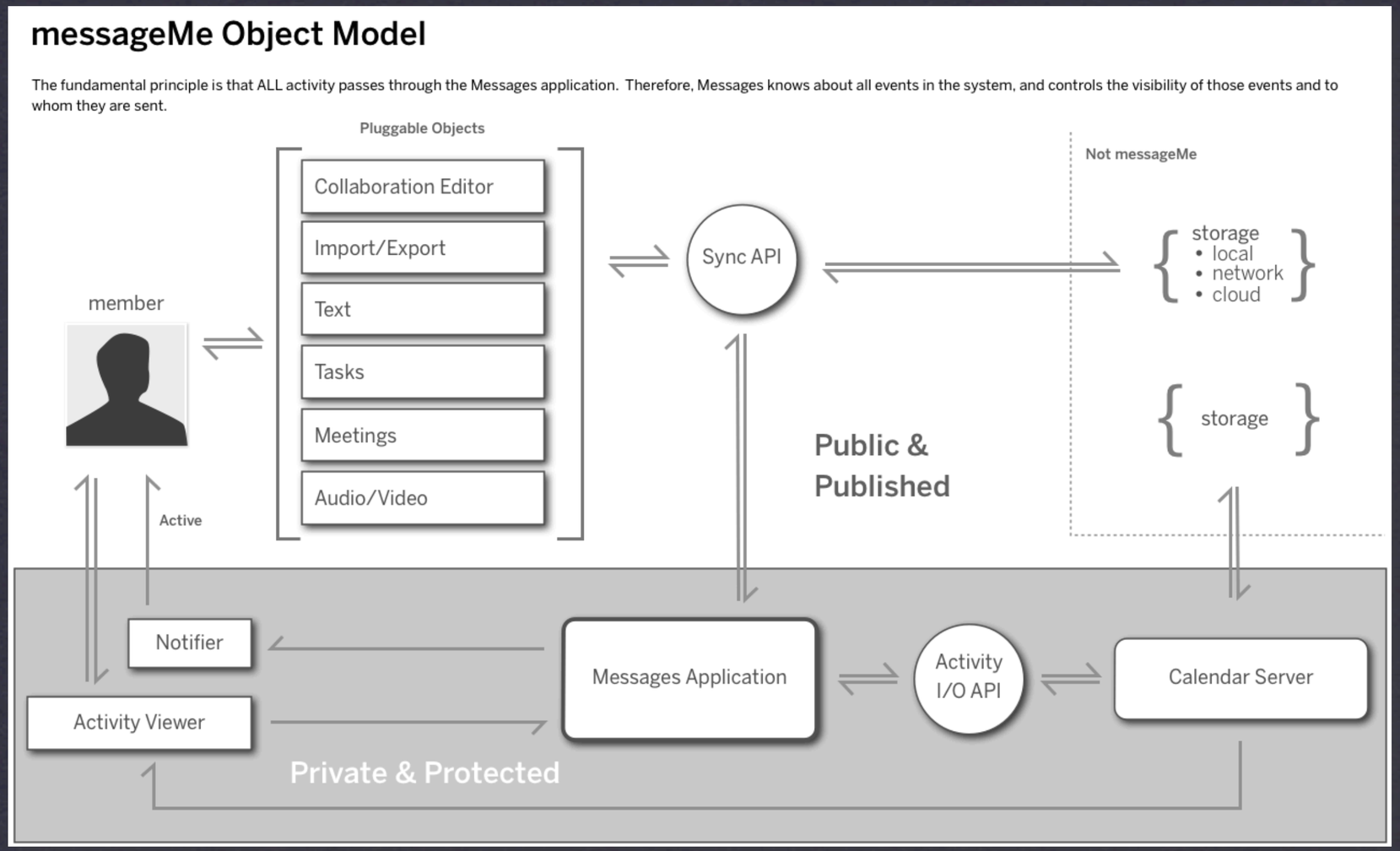
Table

Effective and clear method for communicating object specifics.

Not best for communicating narrative, subtlety or object connections.

*redrawn from Johnson & Henderson, Conceptual Models, 2012, p. 35

Object Model Example



Illustration

The best way to communicate the flow of currency within or the narrative of the system.

Unless extensively annotated, requires some prose-based supplement to define the object details.

Object Modeling Challenges

- **Level of modularity**

  objects contain objects, what is the right level of detail to consider for the object model?

- **When to group, when to separate, how much to distinguish**

  are two things different because they are different objects or different instances of the same object and then how many types of an object are there?

- **User invisible objects and non-interactive objects**

  tracking objects that are not part of the view, but are relevant to the model and controller

fin