If data is required for a route, such as node data or the user's favorites, the route requests the data prior to creating the page.
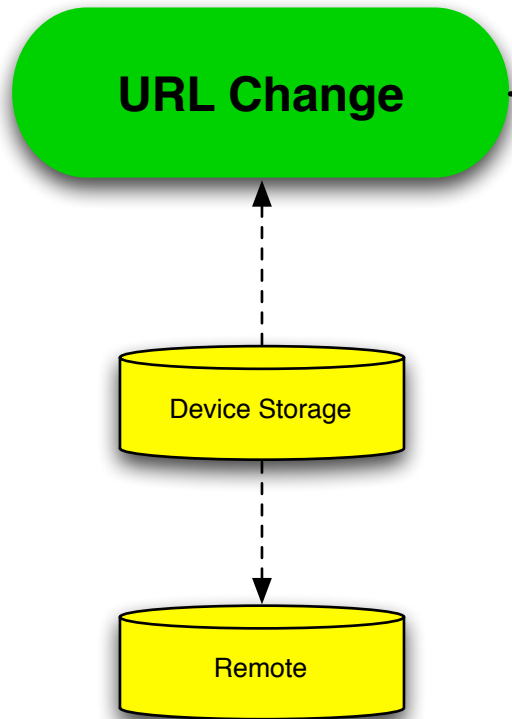
Data

When the app is booted from a "cold" state (that is, it's not fast-app-switched), it goes through a bootstrapping process. This process ensures the application is working with the most recent data, and also sets up various application-wide functionality, including the Router, UI, Page Factory, Data Store, and several others.

Once these pieces are in place, the app triggers a URL change to the home node, and the process outlined here begins.
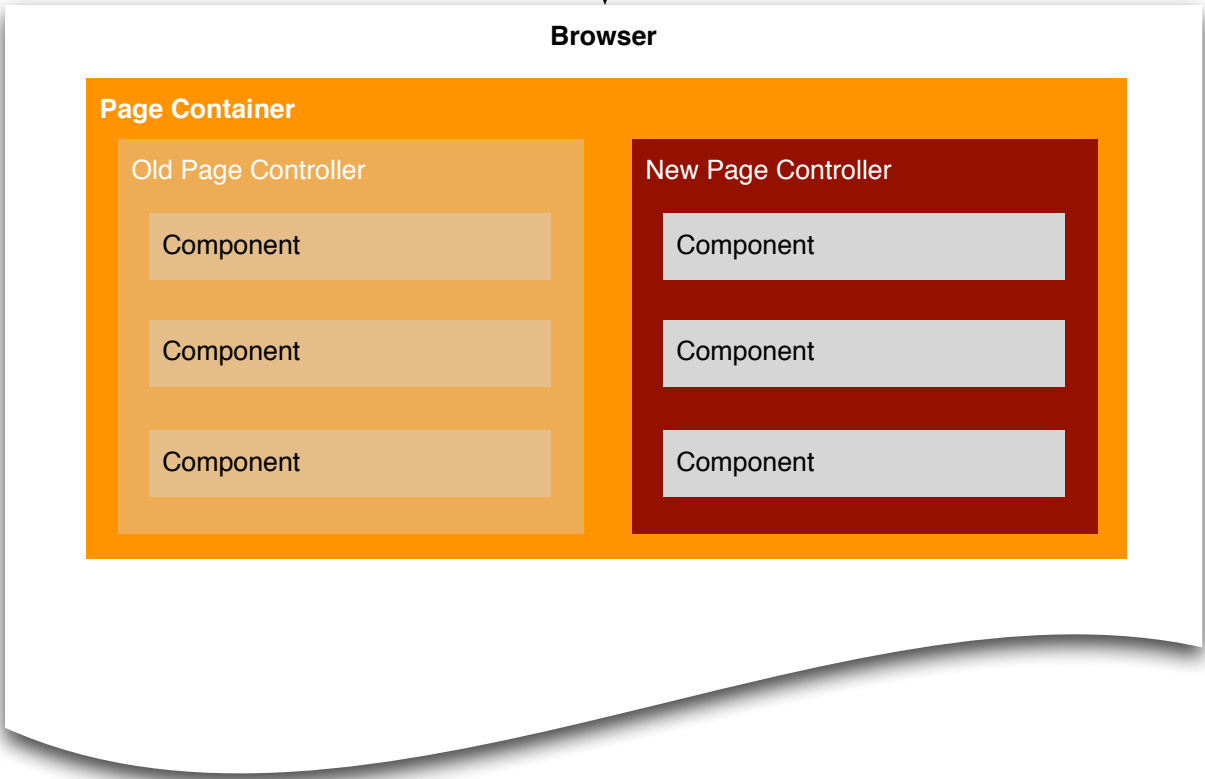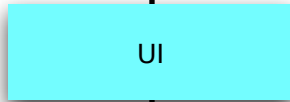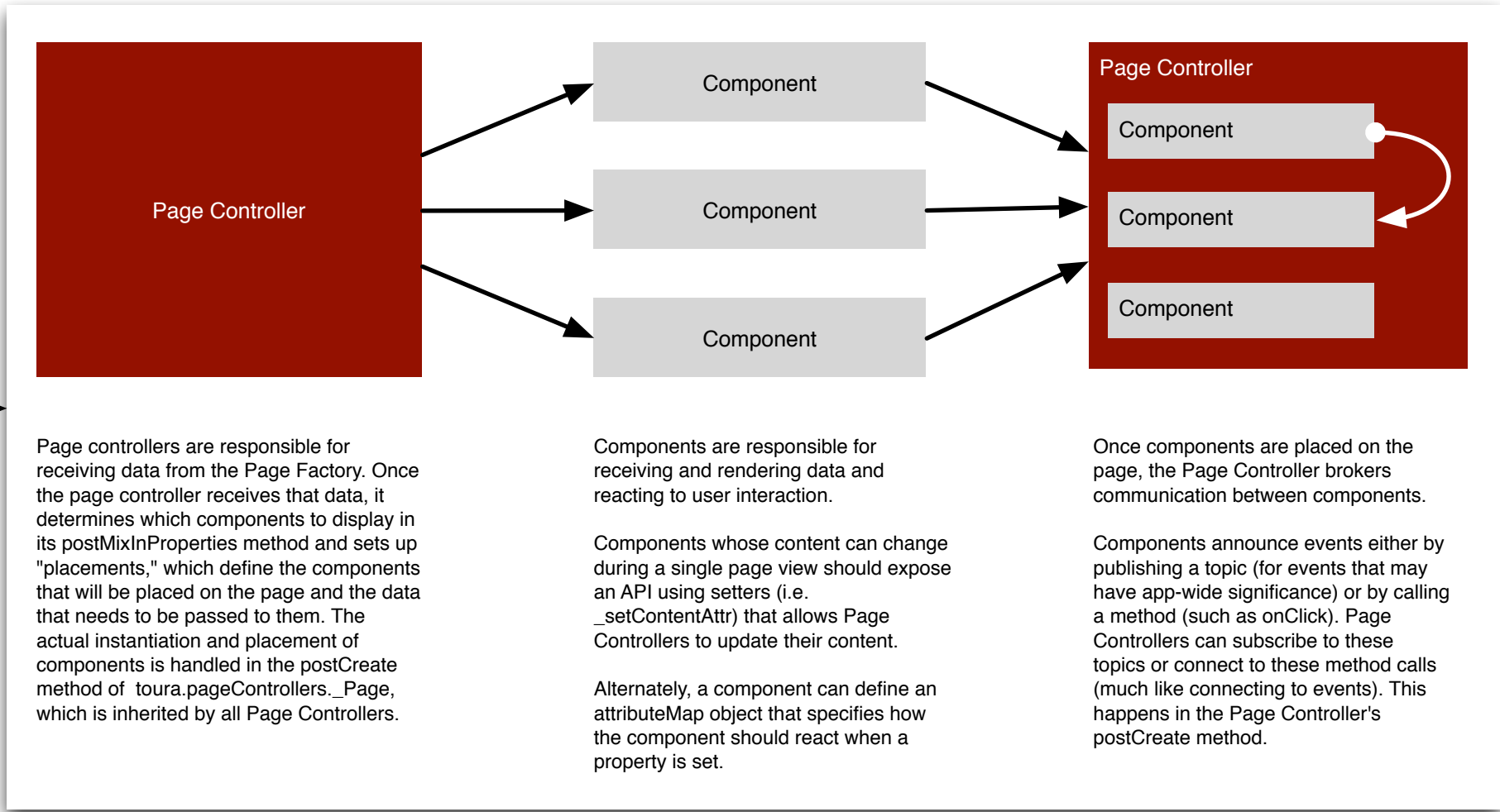
URL Change

Router

Every time the user goes to a new page, we check the age of the local data; if it is more than 8 hours old, we see whether we need to do an over-the-air update.

Device Storage

Remote

The Router detects a change in the URL, looks for a matching route in toura.app.Routes, and parses any additional parameters out of the URL (such as the node ID, asset type, etc.).

Route

Page Factory

The Route asks the page factory to generate a page controller. If the page controller is for a node, the Page Factory figures out which template (Audios, Images, etc.) the node uses. The page factory creates an instance of the page controller and hands the created instance back to the Route.

Page Controller

Component

Component

Component

Page Controller

Component

Component

Component

Page controllers are responsible for receiving data from the Page Factory. Once the page controller receives that data, it determines which components to display in its postMixInProperties method and sets up "placements," which define the components that will be placed on the page and the data that needs to be passed to them. The actual instantiation and placement of components is handled in the postCreate method of toura.pageControllers._Page, which is inherited by all Page Controllers.

Components are responsible for receiving and rendering data and reacting to user interaction.

Components whose content can change during a single page view should expose an API using setters (i.e. _setContentAttr) that allows Page Controllers to update their content.

Alternately, a component can define an attributeMap object that specifies how the component should react when a property is set.

Once components are placed on the page, the Page Controller brokers communication between components.

Components announce events either by publishing a topic (for events that may have app-wide significance) or by calling a method (such as onClick). Page Controllers can subscribe to these topics or connect to these method calls (much like connecting to events). This happens in the Page Controller's postCreate method.

UI

Browser

Page Container

Old Page Controller

Component

Component

Component

New Page Controller

Component

Component

Component

The Route asks toura.app.UI to place the Page Controller in the UI; in turn, toura.app.UI sets the content attribute of the Page Container. If there is already a page on screen, the Page Container handles the animation between the pages, and calls the destroy method of the old page, which results in the proper teardown of the old page and its components.