website, whether generated by a top-notch research team on staff or contributed by users around the world. Data motivates the product that users enjoy, so architects build the rest of the traditional "n-tier" software stack (the *logic* and *display*) around it.

This is the story of Facebook's data and how it has evolved with the creation of the Facebook Platform.

Facebook (*http://facebook.com*) stands as an example of an architecture built around data of great utility, including user-submitted personal relationship mappings, biographical information, and text or other media content. Facebook's engineers built the rest of the site's architecture with an eye to displaying and manipulating this social data. Most of the site's business logic closely depends on this social data, such as the flow and access patterns of various pages, implementation of search, surfacing of News Feed content, and application of visibility rules to content. To the user, the value of the site springs directly from the value of the data he and his social connections have contributed to the system.

"Facebook the social website" is conceptually a standard n-tier stack, where a user's request fetches data from Facebook's internal libraries, which is then transformed through Facebook's logic, to be output through Facebook's display. Facebook's engineers then recognized the usefulness of this data beyond the confines of its container. The creation of the Facebook Platform markedly changed the gestalt of Facebook's data access system, accommodating a vision much broader than the isolated functionality of the n-tier stack, to integrate outside systems in the form of *applications*. With the user's social data at the center of the architecture, the Platform has developed into an array of web services (Facebook Platform Application Programming Interface, or Facebook API), a query language (Facebook Query Language, or FQL), and a data-driven markup language (Facebook Markup Language, or FBML) to wed application developers' systems with Facebook's.

As given sets of data become more widely available and users demand unified use of their data across multiple web and desktop products, the architect reading this chapter will likely find herself either a consumer of such a platform or the producer of a similar platform surrounding her own site's data. This chapter takes the reader on the journey of opening up Facebook's data to outside stacks in a controlled way, the architectural choices that follow from each step of the data evolution, and the process of reconciling this with the unique privacy requirements that permeate social systems. It includes:

- Motivating scenarios for these types of integrations
- Moving data functions from an internal stack call to an externally visible web service (the Facebook API)
- Authorizing access to this web service with an eye to maintaining the privacy of the social system
- Creating a data query language to ease the burden of new clients of this web service (Facebook FQL)