App-enabled EHR RFP

For your EHR vendor to support applications (apps) they should design products that make it possible for third-party developers to build apps and services that tightly integrate with your EHR. Fast Healthcare Interoperability Resources (FHIR) is an emerging technology that supports this approach. Many platforms, including SMART, use FHIR to improve the usability, interoperability, and customization of health information technology (health IT).

The SMART on FHIR technical requirements detailed below can serve as the basis for getting the most out of your new EHR. You or your IT staff may wish to augment this list of minimum requirements with suggestions that more closely fit you and your organization’s needs. At a minimum, the vendor product should include the following components in order to support app-enabled applications:

Data Access

* Provide automated, standards-based, read-only access to:
	+ A well-defined set of real-time discrete data (represented in FHIR, with appropriate profiles and vocabularies)
	+ Automated bulk export of standards-based data

Data Manipulation

* Write structured data from third-party apps back to the organization’s EHR and where relevant, a data warehouse, using the FHIR REST application programing interfaces (API) to communicate data including:
	+ Free-text clinical notes

Standards-Based App Authorization

* Protect data and identity endpoints with standards-based authorization mechanisms (OAuth2)

Identity Management

* Act as a standards-based Identity Provider using OpenID Connect. This ensures that users can authenticate to plug-in apps using single-sign-in via their existing EHR credentials.
* Act as a standards-based relying party to a customer-selected Identity Provider using OpenID Connect. This ensures that users can sign into the EHR using an external, hospital-supplied single-sign-on account.

Workflow

* Support standards-based embedding of external application user interface (HTML5). This ensures that app developers can build Web apps, and these apps can run directly inside of the EHR.
* Support the launch of external applications in the clinician’s workflow (this is not limited to the EHR, and should include non-EHR integrated tools such as smart phones and tablets). For example, a clinician that has opted to use a third-party-developed native iPad app to visualize a patient’s BMI over time can seamlessly use the application alongside the EHR via single-sign-on.
* Support notifications to and from running applications. For example, an embedded app can notify the EHR when the user is “done” with it.

Add-On Functionality

Your organization may also want to consider the following additions to its RFP depending on the types of applications it wishes to develop and run in the future.

Data Access

* Provide automated access to bulk export of vendor-specific data (complete representation of all data).

Data Manipulation

* Write structured data from third-party apps back to the organization’s EHR and where relevant, a data warehouse, using the FHIR REST API to communicate data including:
	+ medication prescriptions
	+ lab and diagnostic imaging orders
* Support the dependent transactions necessary to ensure that actions completed by third-party applications using the API are valid in the EHR and data warehouse.

Standards-based App Registration

* Support Auth Dynamic Registration (with access controlled by the customer) so that third party apps can perform automated, standards-based registration.

Context-Specific Service Hooks

* Support the ability to call an external standards-based service in specific workflow steps, such as:
	+ new prescriptions
	+ new lab orders
	+ new imaging studies

Intellectual Property

The intellectual property (IP) of any app integrated through the SMART on FHIR API belongs to the author and not the vendor.

Custom SMART on FHIR Extension to a Proprietary API

Should a vendor neglect to provide SMART on FHIR natively, the client has the right to provide a custom extension to the vendor’s API. The ownership of the IP for the custom extension is negotiable between the client and the vendor, but the ownership of the app using the custom extension belongs to its author.

*Source: AMA. Practice transformation series: revenue cycle management in medical practice. 2015.*