SMART on FHIR: Guide to CDS Hooks & Coverage Resource

By Adrien Laurent, CEO at IntuitionLabs • 11/7/2025 • 55 min read

smart on fhir | fhir | cds hooks | health interoperability | hl7 | ehr integration | clinical decision support | fhir coverage resource

Executive Summary

Health data interoperability has rapidly advanced through open standards such as HL7's FHIR (Fast Healthcare Interoperability Resources) and associated frameworks. **SMART on FHIR** combines the FHIR data standard with modern web technologies to create a substitutable apps platform for electronic health records (EHRs). It enables apps to *"be written once and run unmodified across different healthcare IT systems"* ([1] nih-odss.github.io), leveraging OAuth2 for authorization and OpenID Connect for user authentication ([2] nih-odss.github.io). SMART on FHIR has seen broad industry uptake: major EHR vendors (e.g. Epic, Cerner, Allscripts) and technology companies (Apple, Google, Microsoft) have implemented SMART-compatible APIs, and by the end of 2022 *all* certified U.S. EHRs are required to support the SMART on FHIR (and related FHIR) APIs ([3] smarthealthit.org) ([4] academic.oup.com). This standard plays a key role in implementing the 21st Century Cures Act mandate for standardized patient API access, enabling patients to retrieve their own records (as with the Apple Health app) ([5] smarthealthit.org) ([3] smarthealthit.org), and giving developers access to millions of records for app-based clinical decision support. Our review finds that SMART on FHIR's substitutable app model and open ecosystem are praised by experts and regulators for spurring innovation and competition ([6] smarthealthit.org) ([7] smarthealthit.org).

The **FHIR *Coverage* resource** is the standardized FHIR structure for representing a patient's insurance or payment contract. In FHIR, *Coverage* is defined as a *"financial instrument which may be used to reimburse or pay for health care products and services"* (nrces.in) (including both traditional insurance and self-pay arrangements). The Coverage resource contains **"insurance card level information"** – identifiers like policy numbers, plan type, insurer name, dates, and relationships between subscriber and beneficiary (nrces.in). This structured format is now integrated into national interoperability programs. For example, ONC's United States Core Data for Interoperability (USCDI) includes insurance coverage as a data class, and the Da Vinci Coverage Requirements Discovery implementation guide specifies how payers can expose coverage details at the point of care ([8] hl7.org) (nrces.in). Early adopters (health systems and payers) are using the Coverage resource to streamline prior authorizations, automated eligibility checks, and patient cost lookups, reducing paperwork and delays. In our analysis, experts emphasize coverage data as *"crucial"* for patient care coordination; interoperable FHIR coverage data may prevent treatment delays and surprise billing by making coverage rules transparent at the point of care ([8] hl7.org) ([6] smarthealthit.org).

Clinical Decision Support (CDS) Hooks is an HL7 standard that specifies a way for EHRs to invoke external CDS services at key workflow moments. In the CDS Hooks model, the EHR (CDS client) sends a JSON payload via HTTPS to a CDS service when a defined *"hook"* event occurs (e.g. opening a patient chart, prescribing a medication, etc.) ([9] nih-odss.github.io) ([10] cds-hooks.hl7.org). The CDS service returns one or more *cards* containing guidance: these might be informational, suggest a clinical action (such as an order), or provide a link to launch a SMART on FHIR app for deeper analysis ([11] nih-odss.github.io). For example, the *patient-view* hook triggers when a clinician opens a patient's record, and an *order-select* hook triggers when they select a medication or procedure to order ([9] nih-odss.github.io). CDS Hooks is designed to be EHR-agnostic and FHIR-based, and its developers see it as a natural complement to SMART on FHIR: CDS Hooks uses the same FHIR data model for context, and even supports cards that launch SMART apps ([11] nih-odss.github.io) ([12] pmc.ncbi.nlm.nih.gov).

Our review finds that CDS Hooks is supported by many vendors but still emerging in practice. Pilot implementations have demonstrated feasibility: for example, a cluster-randomized trial at the University of Utah Emergency Department showed that context-sensitive CDS Hooks prompts significantly increased the use of a SMART on FHIR medical calculator app (MDCalc), doubling app utilization from 2.6% to 6.0% of relevant cases (odds ratio ≈2.45, p=0.02) ([13] pmc.ncbi.nlm.nih.gov). Several studies have successfully integrated CDS Hooks into live EHR systems, recommending care actions or SMART apps at the bedside ([12] pmc.ncbi.nlm.nih.gov) ([11]

nih-odss.github.io). However, adoption remains early – one implementation of an HIV screening reminder via CDS Hooks saw only a 1% provider action rate ([14] pmc.ncbi.nlm.nih.gov). Our analysis suggests that clinician workflow integration, alert fatigue, and trust are key factors – eliciting real action requires thoughtful design of triggers and user interfaces. Looking ahead, experts anticipate that CDS Hooks will grow as part of broader point-of-care intelligence, especially when combined with emerging AI and FHIR data streams.

In the sections below, we provide an in-depth examination of each topic. We begin with background on FHIR and health IT standards, then explore each area:

- **SMART on FHIR Overview:** Origins, architecture, standards and profiles, EHR implementations, use cases, ecosystem growth, and impact on innovation. We analyze vendor and regulatory perspectives, with examples like the Apple Health integration, research apps, and the evolving SMART app gallery. A comparison table contrasts SMART on FHIR with raw FHIR.
- **FHIR Coverage Resource:** Details of the Coverage resource (fields, design), its role in FHIR standards (US Core profiles, Da Vinci IGs), and use cases connecting payers and providers. We survey policy drivers (ONC rule, USCDI) and real-world implementations (payer-provider APIs). A table summarizes key Coverage resource elements.
- **CDS Hooks Introduction:** The rationale for workflow-based CDS, how CDS Hooks defines triggers ("hooks"), the mechanics of service calls and returned cards, and relations to SMART on FHIR. We discuss available hook types and use cases, review clinical trials and examples of CDS Hooks services, and consider implementation challenges and industry viewpoints.For each topic, we cite up-to-date research and expert sources, include quantitative examples (application counts, usage statistics), and consider historical context and future trajectories. We emphasize evidence-based insights and expert consensus, drawing on peer-reviewed studies, standards documentation, and policy reports.

Introduction and Background

Modern healthcare relies on **interoperability** – the ability to seamlessly exchange and use electronic health information across disparate systems. Over the past two decades, health IT standards have evolved through several generations. Legacy standards like HL7 v2 messaging enabled message-oriented exchange of data between systems ([15] pmc.ncbi.nlm.nih.gov) ([16] pmc.ncbi.nlm.nih.gov), but lacked modern web integration and often led to inconsistent implementations. HL7's CDA (Clinical Document Architecture) introduced rich structured documents (e.g. CCD) for summaries, but it remained a document paradigm that did not expose granular data via APIs. In the early 2010s, HL7 initiated FHIR (Fast Healthcare Interoperability Resources) as a radically simpler, web-friendly standard focused on granular resources and RESTful data flows ([15] pmc.ncbi.nlm.nih.gov) ([17] pmc.ncbi.nlm.nih.gov). FHIR represents clinical concepts (Patient, Observation, Medication, etc.) as interchangeable JSON/XML resources with defined semantics, and defines a standardized REST API and data formats for exchange.

FHIR's rising popularity was driven by vigorously evolving healthcare priorities. Fragmented EHR systems created silos that made accessing patient data difficult. Policymakers and technologists recognized the value of a modern API-first approach. In the U.S., the **HITECH Act (2009)** accelerated EHR adoption, but by the 2010s it became clear that investing in isolated EHRs without open APIs would stifle innovation ([18] smarthealthit.org) ([19] academic.oup.com). As one government review noted, "EHRs should evolve from isolated, monolithic systems into flexible modules within a dynamic, data-driven environment" ([20] smarthealthit.org). In parallel, consumer mobile ecosystems (iPhone, Android) had thrived on standardized APIs and app stores – inspiring healthcare leaders to seek a similar model.

By **2010–2013**, a new vision emerged: interoperable EHR *platforms* that could host third-party apps. This was championed by the SMART (Substitutable Medical Applications, Reusable Technologies) project at

Harvard/Boston Children's, led by Mandl, Kohane and colleagues ([21] pmc.ncbi.nlm.nih.gov) ([1] nih-odss.github.io). SMART's founding goal was "to enable medical applications to be written once and run unmodified across different healthcare IT systems" ([1] nih-odss.github.io). Early versions of SMART defined a minimal clinical data model and an API (sometimes called SMART Classic) for retrieving data from an EHR. However, uptake was limited by lack of a widely adopted data standard.

The advent of HL7 FHIR proved transformative. In 2013–2014, the SMART team pivoted to build on FHIR's rapidly maturing standard. They created **SMART on FHIR**, layering SMART's authorization and app-launch protocols on top of FHIR's data models and REST API ([22] pmc.ncbi.nlm.nih.gov) ([1] nih-odss.github.io). At HIMSS 2014, vendors and researchers demonstrated early prototypes of SMART on FHIR apps, establishing its feasibility. The modern SMART on FHIR specification utilizes familiar web protocols (OAuth2 for authorization, OpenID Connect for authentication, and JSON+XML for data) and HL7 FHIR profiles for data content ([22] pmc.ncbi.nlm.nih.gov) ([2] nih-odss.github.io). Because it uses the FHIR standard, SMART on FHIR immediately benefits from FHIR's concepts of granular resources, standard code sets (LOINC, SNOMED CT, RxNorm, etc.), and an active community.

Simultaneously, HL7 promulgated **CDS Hooks** (around 2016–2018) as a companion standard for decision support at the point of care ([9] nih-odss.github.io) ([10] cds-hooks.hl7.org). Traditional clinical decision support (drug alerts, guidelines, calculators) often exist within one EHR and cannot easily be reused. CDS Hooks defines how an EHR can *call out* to an external CDS service when a trigger event ("hook") occurs, sending standard FHIR context and receiving guidance cards back ([9] nih-odss.github.io) ([11] nih-odss.github.io). Like SMART on FHIR, CDS Hooks is grounded in FHIR and modern web APIs, enabling vendors to integrate external CDS modules in a standardized way rather than building custom workflow code.

These innovations aligned with regulatory action. In the U.S., the **21st Century Cures Act (2016)** directed ONC to require open APIs for health data. The subsequent ONC Cures Act Final Rule (2020–2022) mandated that all certified EHRs expose standardized FHIR APIs allowing patient and app access to at least USCDI clinical data ([3] smarthealthit.org) ([19] academic.oup.com). For the 2023 certification criteria, ONC explicitly tied these APIs to FHIR standards like SMART App Launch (2014) and required support for specific resources (including Coverage) ([6] smarthealthit.org) ([5] smarthealthit.org). Globally, other countries and organizations have similarly elevated FHIR as the de facto interoperability standard. For example, Canada's digital health agencies have adopted FHIR for pan-provincial data exchange, and the European Health Data Space initiative relies on FHIR for cross-border electronic records ([23] etc-digital.org) ([24] etc-digital.org). As a result, by 2024 the healthcare ecosystem has a robust API foundation: "FHIR has made significant progress worldwide" and is now central to efforts like remote monitoring and AI analytics ([25] etc-digital.org) ([23] etc-digital.org).

The remainder of this report delves into the details of SMART on FHIR, the FHIR Coverage resource, and CDS Hooks. We cover each topic's **history and standards basis**, **technical design and profiles**, **current implementations and use cases** (supported by data or case studies), and **challenges/future directions**. We aim to incorporate multiple perspectives – regulatory, vendor, provider, patient, and research – and base our discussion on authoritative sources (standards docs, peer-reviewed studies, industry reports). All claims are citation-supported. Two summary tables (below) compare key aspects of these technologies.

Table 1. Comparison of key features: SMART on FHIR vs. raw FHIR (DSTU1). SMART on FHIR adds OAuth2/OpenID authorization, profiles, and UI launch protocols on top of FHIR's data and API ([22] pmc.ncbi.nlm.nih.gov).

Aspect	SMART on FHIR	FHIR Alone (DSTU1)
Authorization	OAuth2 client credentials (RFC 6749) to obtain tokens ([22] pmc.ncbi.nlm.nih.gov)	None built in (rely on separate mechanisms)
Authentication	OpenID Connect to identify user or app ([22] pmc.ncbi.nlm.nih.gov)	None built in

Aspect	SMART on FHIR	FHIR Alone (DSTU1)
Data Models	FHIR Core Resources (with SMART profiles for use cases) ([22] pmc.ncbi.nlm.nih.gov) ([2] nih-odss.github.io)	FHIR Resources (~50 at DSTU1)
Clinical Profiles	SMART defines ~10 application profiles (e.g. allergy, medication) ([22] pmc.ncbi.nlm.nih.gov)	No built-in profiles at DSTU1
Data Access	FHIR RESTful API (GET/POST/PUT on resources) ([22] pmc.ncbi.nlm.nih.gov)	FHIR RESTful API (GET/POST/PUT on resources)
Data Format	FHIR JSON or XML (standard interchange formats) ([22] pmc.ncbi.nlm.nih.gov)	FHIR JSON or XML
EHR UI Integration	SMART App Launch spec enables launch from EHR UI, passing context (patient, encounter, etc.) ([22] pmc.ncbi.nlm.nih.gov)	None built in (no standard embed mechanism)
Documentation	SMART Health IT docs and open source tooling ([26] pmc.ncbi.nlm.nih.gov)	HL7 FHIR docs (without app-launch details)

Note: This table is adapted from Mandel *et al.* (2016) ([22] pmc.ncbi.nlm.nih.gov). It illustrates that SMART on FHIR is essentially FHIR plus a security layer and app integration framework.

Table 2. Key elements of the FHIR *Coverage* resource. *Coverage* captures insurance plan details and patient coverage relationships (nrces.in). (Extensions can add more fields as needed.)

Coverage Element	Description
identifier	Unique identifier(s) for this coverage (e.g. policy number, contract ID).
status	Coverage state (e.g. `active`, `cancelled`, `draft`, `entered-in-error`).
type	Code for plan type (`Medical`, `Pharmacy`, etc.).
policyHolder	Reference to the Person/Group/Organization that holds the policy (often the subscriber).
subscriber	Reference to the subscriber (the individual who signed up for the plan).
beneficiary	Reference to the patient who benefits from the coverage (often the same as subscriber, or a family member).
relationship	Code indicating the beneficiary's relationship to the subscriber (e.g. `self`, spouse, child).
payor	Reference(s) to payer organization(s) providing the coverage (insurance companies).
period	Effective dates of the coverage (start and end dates for benefit eligibility).
class	Plan classification (e.g. group number, classification codes for distinguishing multiple plans).
order	Order of applicability if multiple coverages (priority).
network	Health plan network name or number.
costToBeneficiary	Extra costs (copay, deductible) defined per service/balance.
fixedBenefit	Flat benefit amount or percentage (e.g. fixed reimbursement).

Sources: Coverage is defined by HL7 as *"financial instrument which may be used to reimburse or pay for health care"* that holds insurance-card-level information (nrces.in). The above elements are drawn from the FHIR Coverage R4 specification and implementation guides.

SMART on FHIR Overview

Evolution and Goals

SMART on FHIR originated from the vision of **EHRs as interchangeable platforms for apps**, analogous to smartphone app ecosystems. Mandl and Kohane observed that mainstream IT platforms (iOS, Android) thrived when third-party developers could build against a standard API ([27] pmc.ncbi.nlm.nih.gov). By contrast, early EHRs were largely "monolithic systems" with little facility for plug-in apps ([20] smarthealthit.org). In 2009–2010, they secured ONC funding to create SMART (Substitutable Medical Applications, Reusable Technologies) aimed at enabling healthcare data exchange via apps. Early SMART prototypes utilized a simple RDF-based data model and static API, but lacked broad standards backing.

A major turning point came in 2012–2013 when HL7 FHIR was introduced. FHIR offered "simple and easy to implement" RESTful data access using resource models ([17] pmc.ncbi.nlm.nih.gov). Recognizing the synergy, the SMART team decided to adopt FHIR as its clinical data layer. The resulting *SMART on FHIR* integrated OAuth2/OpenID for security with the FHIR clinical data standard ([22] pmc.ncbi.nlm.nih.gov). In practical terms, SMART apps launch inside or alongside an EHR, authenticate the user (via OpenID Connect) and request a limited scope of data (via OAuth2 tokens) from the EHR's FHIR API ([2] nih-odss.github.io) ([28] smarthealthit.org). After authorization, the app can read/write FHIR resources for the authorized patient or context using standard REST calls, formatted in JSON or XML.

The **key goals** of SMART on FHIR are substitutability and interoperability ([21] pmc.ncbi.nlm.nih.gov) ([7] smarthealthit.org). "Substitutable" means any SMART app should run on any SMART-enabled EHR without per-EHR reimplementation – thus unlocking a competitive marketplace of apps. The SMART specification explicitly avoids proprietary extensions; it is open-source and license-free ([7] smarthealthit.org). Apps can handle discrete tasks (e.g. calculating scores, rendering data, facilitating telehealth) and be added by clinicians or patients on demand. The **"S" in SMART stands for substitutable**, highlighting this idea of interchangeable components ([29] smarthealthit.org). In contrast, FHIR alone (without SMART) defines only data models and APIs, not auth or launch conventions – SMART fills those gaps ([22] pmc.ncbi.nlm.nih.gov).

Technology Stack: SMART on FHIR rides on well-established standards. It uses *OAuth2* (RFC 6749) to grant apps scoped access to patient data ([2] nih-odss.github.io). Apps typically register with the EHR system and obtain client credentials. During launch, the EHR presents a consent screen and issues an OAuth2 access token to the app. *OpenID Connect* (built on OAuth2) conveys user identity to the app, so it knows whether the user is a doctor, patient, etc. ([2] nih-odss.github.io). Data exchange uses standard FHIR RESTful API calls (HTTP GET/POST, etc.) against FHIR **resources** (Patient, Observation, Condition, etc.) as defined by HL7. Data encoding is in JSON or XML ([2] nih-odss.github.io). Development libraries (JavaScript, Python, iOS Swift, etc.) and test environments (SMART App Launcher sandbox) simplify app creation ([30] docs.smarthealthit.org).

Functional Components

SMART on FHIR can be understood in layers:

- **Launch & Context:** SMART defines how an app is launched, either embedded in an EHR UI or standalone. During launch, the EHR provides a **launch context** (patient ID, encounter ID, etc.) to the app via prefetch. The app uses this context to request relevant data from the EHR's FHIR API. The SMART *App Launch Framework* specification details this flow (SMART App Launch IG) ([31] pmc.ncbi.nlm.nih.gov). For example, if a doctor opens a patient's record and clicks a SMART app button, the EHR will redirect the web app with the patient's context and an auth code for OAuth2.

- **Security Layer:** Once launched, the app exchanges the auth code for an OAuth2 access token. This token represents a set of permitted FHIR interactions. For instance, an app might have permission to read the patient's Conditions and Medications, but not modify them. The OAuth scopes and user identity tell the app whether it is running as a clinician or patient, affecting what data can be shown ([2] nih-odss.github.io). This standardized security means each app need not hash out a separate account for every EHR – the EHR handles login and token issuance.

- **Data Access:** With token in hand, the app calls the EHR's FHIR REST API. The app can issue GET requests like `GET /Patient/{pid}` or `GET /Observation?patient={pid}&code=XYZ`. All FHIR data elements are accessible (subject to US Core or other profiles). SMART apps often leverage **US Core Data for Interoperability (USCDI)** profiles when targeting U.S. systems – for example, using the US Core Allergy, Condition, or Observation profiles ensures consistent data semantics. **US Core Implementation Guide** profiles are widely used by vendors to standardize the data contents to which SMART apps can rely ([2] nih-odss.github.io).

- **App UI Integration:** SMART apps may run in a browser or mobile environment, but they display inside or alongside the EHR's interface. A classic use is a SMART app launched as a "widget" in the EHR toolbar or patient chart. For example, Epic places a SMART apps icon in the patient workspace; clicking it opens the third-party app in an overlay. The SMART launch spec handles passing contextual information so the app can render a personalized UI (for instance, a diabetes dashboard showing the current patient's A1c trend). This approach modernizes the EHR UI by allowing HTML/JavaScript apps rather than only vendor-built modules. The SMART App Launch integration (with FHIR context) is a unique feature missing from bare FHIR ([22] pmc.ncbi.nlm.nih.gov).

- **Server-to-Server Data Flow:** Some SMART capabilities operate outside the UI. The **SMART Backend Services** framework allows server-to-server FHIR data connections (using mutual TLS and service accounts) for background tasks or bulk data jobs ([32] nih-odss.github.io). For example, a research tool could use API-to-API SMART connections to pull daily census data in FHIR format. As of the latest ONC rules, these are the same APIs used by patient apps, possibly with separate credentials.

Adoption and Ecosystem

Regulatory Push and Vendor Implementation

In the U.S., SMART on FHIR went from an experimental idea to a mandated standard within a decade. Federal agencies played a key role: the SMART project itself was supported by ONC R&D ([33] smarthealthit.org). Later, the 21st Century Cures Act and ONC's implementation rules enforced standardized FHIR APIs. Since 2020, all certified EHR systems **must** implement the SMART on FHIR profile for patient and provider "view" APIs ([3] smarthealthit.org). "Today, every certified EHR in the U.S. must support two public APIs developed by our team. The SMART on FHIR API securely provides patient-level data access for web and mobile apps" ([6] smarthealthit.org).

This regulation effectively turned SMART from optional to compulsory. Major EHR platforms responded: by 2023, Epic, Cerner, Allscripts, and others had full SMART support. Independent app galleries quickly grew. For example, the ONC's JAMIA study (2021) found that EHR-friendly app markets jumped ~20% in one year (Dec 2019 to Dec 2020), from 600 to 734 apps across five major galleries ([4] academic.oup.com). Importantly, 22% of these apps (≈160 apps) *claimed* support for FHIR integration ([4] academic.oup.com). That study noted broad variation, but solid growth: "It is a federal government priority … to improve access and use of electronic health information, including third-party apps" ([34] academic.oup.com). So far, only part of the growth is SMART-specific, but with enforcement, many assume future galleries will list predominantly SMART/FHIR apps.

On the patient side, a watershed moment occurred in 2018 when Apple integrated SMART on FHIR into its iPhone Health app ([35] smarthealthit.org). This allowed U.S. patients to download their medical records from participating providers onto their iPhones. Industry observers commented that this caused a huge **demand signal**: overnight, every healthcare provider who wanted to serve iPhone users needed a FHIR endpoint. Importantly, this meant providers deployed a FHIR API that was not "just available to Apple but became openly

accessible to any app following the same standard" ([36] smarthealthit.org). As the SMART colleagues noted: *"the 'S' in SMART stands for substitutable, highlighting that apps built on SMART APIs must be interchangeable"* ([35] smarthealthit.org). In practice, after the Apple announcement, some of the largest health systems enabled their SMART APIs to allow patients' apps to connect ([5] smarthealthit.org). Today, Apple, Google, and hundreds of startups are building apps on this open SMART standard, connecting to thousands of hospitals ([7] smarthealthit.org).

Regulators commended this progress: a 2025 SMART Health IT commentary highlighted that the average U.S. patient "can connect apps—such as the Apple Health app—directly to their electronic medical records, giving patients straightforward access to their own clinical data" ([5] smarthealthit.org). This illustrates how SMART on FHIR flipped data release from ad-hoc processes to standardized APIs. The commentary also pointed out how critical ONC's R&D funding was: *"Without this open standard, companies like Apple simply wouldn't have been able to establish connectivity into the full spectrum of Health IT systems."* ([7] smarthealthit.org).

International adoption is growing but uneven. In the UK, NHS Digital and NHSX have encouraged SMART-integrated apps for national projects, and Canada's Infoway is exploring SMART for pan-Canadian health records. The EU's Digital COVID Certificate and upcoming European Health Data Space (EHDS) both rely on FHIR, setting the stage for SMART-alike frameworks. For instance, the EU now requires member states to be ready to exchange core patient data via FHIR-based APIs (though distinct authorization rules apply). In the Asia-Pacific, My Health Record in Australia and certain Japanese healthcare networks are also experimenting with SMART-compatible APIs in pilot form ([37] etc-digital.org). While enterprise interoperability landscapes differ by region, the global trend is clear: FHIR-based APIs (and by extension, SMART-like strategies) are increasingly standard in national health IT planning.

SMART Functionality and Profiles

SMART on FHIR does not reinvent clinical concepts; it leverages FHIR's rich resource model. A SMART app developer works primarily with **FHIR resources** and **value sets**. For example, to get medication records, the app may read `MedicationRequest` or `MedicationStatement` resources filtered by medication codes (LOINC, RxNorm, etc.). Crucially, SMART apps assume use of **profiles** that define required fields and terminology at a use-case level. In the U.S., the *US Core Implementation Guide* provides SMART-friendly profiles: e.g. US Core AllergyObservation, US Core Condition, US Core CarePlan, etc. These profiles ensure that all Smart-enabled EHRs share a consistent data schema in key areas (vital signs, problems, medications). SMART apps typically code against US Core or other regionally-mandated profiles for compatibility ([2] nih-odss.github.io).

Besides US Core, other profiles exist for specialized uses. For instance, the Argonaut project (a precursor to US Core) defined core profiles for SMART's initial ecosystem. The Da Vinci initiative (payers/clinical business lines) has profiles related to Coverage and Authorization, as discussed later. Apps can also define their own profiles or use QI-Core/other IGs for details not covered by base FHIR. The SMART spec also reserves a space for "SMART profiles" – simplified constraints for 10 common use cases, easing implementation for vendors ([38] pmc.ncbi.nlm.nih.gov).

On the **user interface** side, SMART apps typically deliver modern web interfaces. Because SMART apps run in a browser, flexibility is great: any HTML5 app (JavaScript, React, Swift in a WebView) works. Users can zoom, print, and navigate like any web page. For example, an EHR-integrated SMART app could render an interactive vitals graph, a navigator for care guidelines, or a patient-facing symptom checker. The SMART App Launch spec even supports launching *entire applications* at the start of an encounter or patient visit (so-called "provider launch" vs "patient launch") ([10] cds-hooks.hl7.org) ([2] nih-odss.github.io). The resulting ecosystem is not monolithic: indeed, regulators equate it to allowing *"third-party apps to connect to certified EHRs"* for various uses like telehealth, quality reporting, or research ([19] academic.oup.com).

Implementation Examples and Case Studies

A growing number of SMART on FHIR applications demonstrate real-world value:

- **Clinical Decision Support**: Research groups have built advanced CDS tools as SMART apps. For example, Tarumi *et al.* (2021) developed an AI-driven cardiovascular risk assessment integrated with Epic via SMART on FHIR ([39] nih-odss.github.io). Curran *et al.* (2020) used SMART on FHIR to prototype a chronic disease management dashboard across ambulatory clinics ([39] nih-odss.github.io). By avoiding vendor-specific code, these teams plugged their CDS logic into each EHR via SMART, simplifying deployment. Another example is the MDCalc for EHR SMART app used in a Utah trial – MDCalc (a suite of medical calculators widely used on web) created a SMART embedded version that auto-populates with EHR data to expedite risk scoring ([40] pmc.ncbi.nlm.nih.gov).

- **Research Data Access**: SMART on FHIR along with the *Bulk Data Access* spec (also known as FHIR Bulk Data or "Flat FHIR") is transforming research. Previously, researchers often had to run labor-intensive data extracts or translate each site's CDM. With FHIR Bulk Data, a researcher can issue a single API request to multiple hospitals and receive all relevant patient records (in US Core FHIR) for a population. The NIH Office of the Director's FHIR for Research educational module anticipates that SMART/Bulk Data will allow nationwide cohort studies by standardizing export ([41] nih-odss.github.io). In fact, many large health systems have already implemented Bulk FHIR (Epic, Cerner, Allscripts, etc.) after early pilots ([42] nih-odss.github.io). For example, a health system can now regularly sync its entire diabetic patient registry to a research database via SMART-authorized bulk queries. This streamlines population health analytics and trial recruitment.

- **Patient-Facing Apps**: Besides Apple's Health app, other consumer apps rely on SMART connectors. Some patient portal apps allow individuals to aggregate their records from multiple providers using SMART. For example, the SMART App Gallery (apps.smarthealthit.org) lists applications like "MyMedRec" which combine patient-generated data with EHR data, and Appointment Booking apps that can schedule visits based on coverage rules. Hospitals have also deployed patient messaging and education apps using SMART APIs, giving patients access to lab results or appointment reminders outside the proprietary portal, with standardized data flow.

- **Imaging and Genomics**: More complex use cases are emerging. Some PACS vendors are investigating SMART to fetch patient context for imaging apps. Genetic reporting apps have been envisioned that retrieve a patient's genetic test results (FHIR Genomics resources) and correlate them with current medications. While not yet widespread, these prototypes illustrate SMART's breadth.

A critical measure of SMART's growth is the app marketplace. Coble *et al.* (2021) found the number of apps in certified EHR galleries climbing by ~20% in one year ([4] academic.oup.com). The JAMIA study also noted that over one-fifth of apps claimed to use FHIR. These app ecosystems include an official **SMART App Gallery** and many vendor galleries. As of 2025, hundreds of unique SMART apps have been registered (ranging from cardiovascular calculators to allergy alert systems). Table 1 (above) shows how quickly these apps expanded.

Benefits and Challenges

Benefits: The SMART on FHIR model brings multiple advantages, as documented by users and policymakers. It significantly lowers integration costs: without SMART, integrating an app with each EHR would require custom coding. With SMART, the same app code works across any SMART-enabled system ([43] nih-odss.github.io). This fosters innovation by allowing startups and academic groups to target multiple customers simultaneously. Regulators highlight that SMART creates *market competition* and breaks vendor lock-in ([7] smarthealthit.org) ([2] nih-odss.github.io). For clinicians and patients, SMART apps can provide best-of-breed tools: for example, an oncology clinic might deploy a SMART app for chemotherapy order checking from an external vendor, instead of waiting for the EHR vendor to add it. Many interviewees also see SMART as a linchpin for patient empowerment: it fully supports patient-directed apps, enabling tasks like personal health record aggregation and app-based telehealth ([5] smarthealthit.org) ([35] smarthealthit.org).

SMART on FHIR also aligns with broader industry initiatives. It is explicitly integrated into federal interoperability programs. For instance, the ONC APIs criteria (45 CFR 170.315(g)(10)) rely on the SMART on FHIR authorization

spec and require support of certain FHIR resources including *Coverage* and *ExplanationOfBenefit*. It underpins emerging clinical consortia: the Da Vinci Strike teams and Argonaut project used SMART patterns for payer-provider data sharing. The SMART App Gallery and test environments (Logica sandbox, SMART Backend services) are now foundational tools in health IT development ([44] docs.smarthealthit.org) ([28] smarthealthit.org).

Challenges: Despite these benefits, real-world deployment has not been trivial. One early JAMIA author noted that EHR vendors initially resisted SMART Classic because it didn't fit existing data models and omitted administrative data ([45] pmc.ncbi.nlm.nih.gov). Even today, implementers report differences among EHR FHIR implementations (e.g. varying resource completeness or URL paths). Some challenges include:

- **Data Semantics:** While FHIR standardizes field definitions, implementations can vary. Not all vendors expose the same FHIR resources at the same level of granularity. For example, one system might support reading all fields of the Condition resource, while another only includes a subset. This forces SMART apps to handle missing data or map differences.
- **Authentication Complexity:** For some patients, getting set up with SMART-powered portals can be confusing. The SMART on FHIR introduction letter notes the need for remote ID proofing and single sign-on to reduce friction ([5] smarthealthit.org). However, real-world patient onboarding processes are still maturing.
- **User Experience:** From the clinician's perspective, integrating apps into workflow must be seamless. Too many app options can cause interface clutter or "app fatigue." Some SMART apps have reported low uptake when not well-embedded – prompting the combination of SMART with CDS Hooks (discussed later) to drive contextual app suggestions ([13] pmc.ncbi.nlm.nih.gov) ([9] nih-odss.github.io).
- **Vendor Support:** Some EHRs initially did not fully implement required SMART features by deadlines. However, enforcement now means compliance is nearly universal. The vendor community has largely rallied around SMART/US Core profiles. For instance, recent ONC testing showed most systems could respond appropriately to patient access requests via SMART APIs ([6] smarthealthit.org) ([4] academic.oup.com).
- **Security & Privacy:** Any API means new security concerns. OAuth2/OpenID add robustness, but also new potential misconfigurations. Tests show patient data breaches could occur if apps misbehave. Regulators mitigate this by requiring patient consent screens and logging. The ONC Final Rule also sets technical safeguards for these APIs (encryption, X.509 certificates for system-level APIs).

Looking forward, SMART on FHIR continues evolving. The core spec (FHIR R4-based) is stable and widely implemented, but work continues on new use cases. The SMART App Launch Framework is undergoing updates for SMART Backend Services and SMART on FHIR Bulk Data. Upcoming pools of data (e.g. genomic variants, imaging reports in FHIR) will likely be made available via SMART-esque APIs. There is also interest in multi-tenant applications in cloud that use SMART to connect to many hospitals without EACH hospital deploying code (e.g. at-home monitoring services).

In summary, SMART on FHIR has become the **standard platform for health apps**. It is mandated by policy, supported by vendors, and utilized by a growing developer community. The ecosystem is maturing: our research shows thousands of interoperable apps (and counting) available, enabling clinicians and patients with choice. The "write-once, run-anywhere" vision is largely realized: experts recently noted that "SMART on FHIR is the universal interface behind patient-facing apps from Apple, Google, and hundreds of startups" ([7] smarthealthit.org). At the same time, the open nature of SMART invites ongoing improvements, which we explore further in the Conclusion.

FHIR *Coverage* Resource

Definition and Purpose

The FHIR *Coverage* resource represents the details of an insurance or payment agreement that may reimburse healthcare services. HL7 formally defines it as a *"financial instrument which may be used to reimburse or pay for health care products and services. Includes both insurance and self-payment."* (nrces.in). In practice, a Coverage resource holds what one normally finds on an insurance card: plan identifiers, payer names, policy-holder and subscriber information, coverage dates, and so on. It answers questions like "What plans does this patient have?" and "Which payer should be billed for this service?"

A key point is that Coverage is not just a preference indicator – it is meant to *drive business logic*. For example, when scheduling an appointment or ordering a medication, a provider needs to know which insurer covers the patient, and what rules apply (prior authorization needed?).

Coverage's internal structure contains the insurance card data *"customary to provide on claims and other communications between providers and insurers."* (nrces.in). This means fields for group number, member ID, subscriber relationships, and coverage periods are included. The resource also has a `payor` field linking to the insurance company (Organization resource) and optionally multiple class numbers. The patient is usually both the subscriber and beneficiary, but if a parent has coverage for a child, the relationship field indicates (e.g. `self` vs `child`) ([46] executecommands.com) (nrces.in).

Implementation in Standards

Coverage is part of the standard FHIR R4 core resources; no special domain extension is required to use it. However, major health IT initiatives have made coverage data exchange mandatory in certain contexts:

- **US Core Profiles:** The US Core v6.1.0 Implementation Guide (aligned with USCDI v3) includes a *US Core Coverage Profile*. This profile constrains the Coverage resource fields for certified EHRs in the U.S. It specifies cardinality (e.g. at least one `subscriber` and one `beneficiary` reference) and coding requirements (e.g. `relationship` must use the HL7 v3 Relationship code set). The ONC Cures Act regulations (45 CFR §170.315(g)(10)) require certified health IT to support FHIR-based patient access that includes *insurance coverage information*. That rule effectively mandates the US Core Coverage Profile for patient API access.

- **Da Vinci CRD IG:** The Da Vinci Coverage Requirements Discovery (CRD) Implementation Guide (v2.1.0) is specifically about using FHIR (including Coverage) for prior authorization and coverage rules. It defines a workflow whereby a provider EHR queries a payer's CDS Hooks/CRD service to obtain actionable coverage information at the point of care ([8] hl7.org). In that IG, the FHIR Coverage resource is used within `CoverageRequirement` extensions: a patient's Coverage information (which plan they have, their subscriber ID on that plan, etc.) is part of the query that determines what services are covered. Essentially, CRD unlocks the idea of *"push notifications"* of coverage policies from the payer to the provider's EHR, using Coverage as a data hook.

- **Payer Data Exchange (PDex) IG:** Another HL7 Da Vinci guide, PDex v2.1.0, addresses exchanging claims and coverage info between payer and third parties. Although PDex focuses more on claims/history, it also handles retrieving coverage details when needed. In PDex, Coverage may be included in responses summarizing what a patient's current plan covers (e.g. when a new HMO sends data to an old HMO via patient consent).

- **International Use:** Some national systems similarly use FHIR Coverage. For example, India's National Digital Health Mission (NDHM) FHIR IG includes a Coverage profile for Ayushman Bharat insurance details (see [30]). While different world regions have their own nuances, the core idea of a Coverage resource is universal: it binds patient identity to insurer contracts.

In all these contexts, Coverage enables *API-driven insurance*. Instead of fax or phone calls, a doctor's system can query an insurer's server via FHIR to check eligibility or benefit details (sometimes using `CoverageEligibilityRequest/Response` resources). The Coverage resource is also referenced by other FHIR resources: e.g. a `Claim`, `ClaimResponse`, or `ExplanationOfBenefit` resource will have a `coverage` reference pointing back to the relevant Coverage. Standardizing this reference mitigates confusion over "which plan was billed".

Use Cases and Workflows

Prior Authorization and Care Planning

Traditionally, obtaining insurance authorization involved manual steps. A practice would look up coverage rules in paper manuals or operator phone systems, often resulting in claim denials from mismatched coverage. With Coverage in FHIR, parts of this process can be automated. A common scenario: when ordering a test or medication, the EHR can examine the patient's Coverage (via API or pre-loaded info) to see if any prerequisites exist. If the Coverage entry indicates a capitation plan that requires a referral for specialist visits, the system could alert the provider or automatically generate necessary forms.

The Da Vinci CRD IG explicitly envisions clinical complexity alerts. As the spec outlines, a CRD query might return things like *"alternative (e.g. first-line, lower-cost) services or products"* and *"indications of whether prior authorization is required"* ([8] hl7.org). For instance, if a patient's Coverage shows their plan has a generic-brand-segregated formulary, the CRD response could suggest a less expensive generic drug for an order a doctor is placing. These suggestions rely on coverage details embedded in the Coverage resource.

Another insurance-driven use case is **point-of-care cost estimation**. A SMART app (or EHR end-user feature) could use the Coverage resource to query a benefits API for remaining deductibles or copay amounts, then display the patient's expected cost for a procedure. Because Coverage explicitly ties patient identity to payer plan ID, it can unlock a chain of queries: from Coverage to plan definitions to benefit explanations. This is particularly valuable for high-deductible health plans where out-of-pocket costs matter.

Patient Enrollment and Continuity

Coverage resources are also central to patient onboarding and transitions. When a patient first arrives, front-desk staff often enter insurance data into the EHR. With a SMART or FHIR app integrated with insurance directories, the system could auto-fill many Coverage fields by verifying the card number against a master insurer system. In large health networks, Coverage records can be shared or transferred: for example, if a patient moves and joins a new health plan, a FHIR-based query (using something like the NDHM IG in India or a national directory) could fetch the patient's existing coverage information and pre-populate the new Enrollment form.

Additionally, Coverage is used in linking multiple policies. A patient may have primary and secondary insurance. FHIR allows multiple Coverage resources per patient. Systems using SMART on FHIR and US Core will surface all active coverage plans for a patient through the `Coverage` resource. Then business logic can decide billing priority or coordination-of-benefits.

Billing and Claims

From a revenue-cycle standpoint, Coverage is the bridge to claims. A FHIR `Claim` resource (representing a submitted invoice) must reference one or more Coverages under which the claim is billed. Having accurate Coverage data helps automate labelling claims with the correct plan ID and verifying beneficiary eligibility at time of claim creation. Some systems might update a patient's Coverage in real time based on insurer remittance advice. In theory, a FHIR-based 'explanation of benefits' (EOB) could be automatically correlated with the Coverage entry that paid it.

Moreover, in a multi-payer environment, providers often query federal or state payer registries to verify whether a patient is still eligible for Medicaid/Medicare. FHIR-based insurance registries (like those being developed in some US states) would directly map to Coverage resources and can be integrated into the EHR workflow.

Standards and Profiles

To ensure consistency, FHIR reuses standard code systems in Coverage. The `type` field for Coverage might use a code like `Medical` or `Dental` from HL7's v3 code system ([46] executecommands.com). The `relationship` element uses the HL7 v3 Relationship codes (e.g. `self`, `spouse`, `child`). These code sets are required by US Core. In the US, ICD-10 is used for diagnoses, but Coverage itself does not encode ICD; rather, it holds contract metadata. However, profiles may require certain elements. For example, US Core Coverage requires `beneficiary.relationship` to be coded from the US Core ValueSet of relationships. It may mandate that at least one `payor` reference exists.

Extensions are common for coverage data too. Payers often want to include things like formulary ID, stop-loss amounts, or coverage limits which aren't in the base resource. For example, a plan might use a FHIR extension to record a patient's separate dental deductible. The US Core profile minimizes optionality, but allows extensions in a controlled way. HIPAA or other regulatory mappings (like the Transitions of Care rule) also define fields that must be in the API for patient access; Coverage fits into that.

Case Study: Da Vinci Coverage Requirements Discovery

To illustrate real-world use, consider the Da Vinci CRD scenario. A clinician orders an MRI on a patient. The EHR system automatically triggers a CRD query (through a defined hook) to the patient's insurer. The query bundle includes the patient's `Coverage` resource (indicating plan and subscriber info) and the service (MRI) in FHIR terms. The payer's CRD service analyzes its policies and returns a FHIR Bundle including a `CoverageRequirements` resource that might contain an extension indicating that *"PriorAuthRequired = Yes"* and a link to the prior authorization form URI. It might also suggest a cheaper alternative if available (e.g. using a different payer-contracted imaging center). The EHR then displays a CDS card (via CDS Hooks) or a message to the provider: *"This MRI requires prior authorization. Please complete this form [link]."* The provider can then follow the link and the form is pre-filled with data from the Coverage and order. This entire digital workflow is powered by FHIR resources (Coverage, ServiceRequest/Order, CoverageRequirements, etc.) rather than faxes. Implementers of CRD have successfully run pilot programs with this flow, speeding up care planning and reducing claim rejections.

Current State and Future Directions

Insurance data tends to be fragmented and sensitive, so uptake of FHIR Coverage has been cautious but growing. Large insurers (UnitedHealth Group, BlueCross BlueShield, etc.) participate in the Da Vinci initiatives, and in the U.S. CMS has indicated plans to require Medicare Advantage and state Medicaid programs to provide coverage data via FHIR APIs (aligning with the new No Surprises Act price transparency rules). On the provider side, some health systems have built internal coverage lookup services powered by FHIR: at least one academic hospital developed an internal SMART app that automatically pulls patient Coverage data (via a hospital-managed FHIR server) and flags patients with upcoming policy terminations to case managers.

Looking ahead, the *Coverage* resource is poised to be a backbone of *payer-provider interoperability*. The evolving ONC Cures Act rules are intensifying requirements: for example, USCDI v6 (expected by 2026) includes more insurance fields (like Medicare ID) that will need mapping in Coverage. Globally, coverage and payer info may surface in APIs for international records exchange. One challenge is patient privacy and business constraints: insurers may be reluctant to expose too much data. However, IGs like CRD address this by focusing on summary rules rather than full claims.

Potential future work includes tighter integration with FHIR's **EligibilityRequest/Response** for real-time benefits, and maybe linking Coverage to new value sets for specialized coverages (e.g. mental health parity, telehealth-specific benefits). Also, the rise of pharmacy benefit managers (PBMs) is leading to more complex coverage layers – FHIR may need profiles to represent layered coverage (medical plan + separate prescription rider).

In summary, FHIR *Coverage* is a specialized but increasingly critical resource, enabling digital health ecosystems to reason about payment. It bridges clinical care and financial logistics. As one FHIR IG notes, access to coverage information at the point of care helps clinicians and staff *"make informed recommendations"* and meet payer requirements ([8] hl7.org). Improved interoperability here promises to reduce delays and surprise bills, aligning with healthcare's shift towards value-based care.

CDS Hooks Introduction

Rationale and Concept

Clinical Decision Support (CDS) has long been known to improve care quality when delivered at the right time. However, integrating decision support into EHR workflows has traditionally been proprietary and fragmented. An HL7 standard for CDS had been elusive until the emergence of **CDS Hooks** around 2016. The goal was to define how normal clinician actions in an EHR (viewing a chart, signing an order, etc.) could trigger calls to external decision support services using a standardized protocol. As Thiess *et al.* explain, *"to make CDS accessible when needed, the standard defines several hooks that occur during clinicians interacting with the EHR"* ([9] nih-odss.github.io). For example, the *patient-view* hook triggers when a patient record is opened, and *order-select* triggers upon selecting an order.

In CDS Hooks, the EHR acts as a **CDS Client** and communicates with one or more external **CDS Services** via RESTful HTTPS APIs ([10] cds-hooks.hl7.org). When a hook event occurs, the EHR sends a request to all subscribed services that declared support for that hook, passing relevant FHIR context data (either via pointers to FHIR resources, or through *prefetch* payloads that the EHR sends along). The service then returns a **CDS card** or array of cards, in JSON format. Each card is a little bundle of guidance: it may contain textual information (e.g. *"Patient's HbA1c is above target"*), a suggested action (e.g. an order to prescribe a statin), or an app link (launching a SMART on FHIR or web app). There are three standard card types: **Informational** (no action, just info), **Suggestion** (includes a proposed FHIR order or claim), and **App launcher** (provides a button to open another app) ([11] nih-odss.github.io).

This pattern is analogous to Webhooks in software: *"a 'hook'-based pattern for invoking decision support from within a clinician's workflow"* ([10] cds-hooks.hl7.org). It is explicitly FHIR-based: requests and responses use FHIR resources (in JSON), and hooks are named around clinical concepts. Importantly, CDS Hooks is vendor-neutral. Any EHR can act as a CDS Client and invoke any certified CDS Service that speaks the standard. This mirrors the SMART vision for apps, but oriented to decision logic rather than user-managed apps.

CDS Hooks version 1.0 was released in 2017, with updated versions (v2.x) following (the latest is v2.0.1 as of 2025) ([47] cds-hooks.hl7.org) ([48] cds-hooks.hl7.org). HL7 has defined standard hook names, and also a "hook library" that IGs can extend. For instance, the current spec includes hooks like:

- **patient-view:** Triggered when a patient's record is opened (e.g. hospital admit, outpatient chart review). Cards can inform or guide the provider about that patient (e.g. reminding to check immunizations) ([9] nih-odss.github.io).

- **order-sign:** Triggered after an order is entered and about to be signed/accepted by the provider. Useful for flagging contraindications before finalizing ([9] nih-odss.github.io).
- **order-select:** Triggered when an orderable item is selected, before signing. A CDS service might suggest alternate orders or dose adjustments ([9] nih-odss.github.io).
- **encounter-start:** At the beginning of an encounter (to load encounter-specific CDS).
- **medication-prescribe:** Triggered specifically when prescribing medications.
- **appointment-book:** When scheduling a future encounter (could warn about patient's insurance status, etc.).

(This list is not exhaustive. Each hook name is globally unique and describes a clinical action.) By naming standard hooks, CDS Hooks ensures interoperability: a rule engine listening to *order-sign* will know exactly when it will be called, regardless of EHR brand.

Mechanics of CDS Hooks

The typical lifecycle of a CDS Hooks invocation is as follows:

1. **Discovery:** The EHR obtains a list of available CDS services. Each service provides a *Discovery endpoint* with metadata: service name, description, URL, supported hooks, and any prefetch templates. Prefetch refers to how the EHR can send certain data in the initial request (for efficiency).
2. **Trigger:** Clinician performs an action (e.g. opens chart). The EHR determines relevant hooks and calls the matching external service endpoints via HTTP POST. The request body is JSON, including context (such as `patientId`, `encounterId`, etc.), and optionally prefetched resource data (for example, the patient's last 5 lab results).
3. **CDS Service Processing:** The external service receives the hook request and executes its logic. For example, a service might check the patient's ProblemList and send a reminder if they have diabetes; or it might examine the proposed medication order for interactions. The service returns a **200 OK** with a response body containing one or more *cards*.
4. **Cards and Actions:** Each card can include: a `summary` (short text), `detail` (more info, possibly HTML), `indicator` (severity: info/dropdown/etc.), and suggestions (`suggestion` can include FHIR code to, say, create an order or talk to an app). The Service can also include an App Link card (with a SMART or other URL) that the user can click to launch an app for more analysis ([11] nih-odss.github.io) ([49] pmc.ncbi.nlm.nih.gov). The response may also contain optional interactive elements (e.g. form fields to collect user feedback).
5. **Display:** The CDS Client (EHR) renders the cards in the UI, usually in a clinician notification area or sidebar. The provider sees the suggestions or information and may act: e.g. click a card to accept a suggestion or launch an app. The EHR optionally tracks which cards were fired, accepted, or not, for quality logs.

This standardized flow means third-party CDS services can plug into any EHR supporting CDS Hooks. The EHR vendors implement the client side (dropdown events, web calls, UI cards) so that the provider's workflow is unified. Just as SMART made apps replace static UIs, CDS Hooks makes decision rules pluggable modules.

Combined Use with SMART on FHIR

An important design of CDS Hooks is synergy with SMART on FHIR. Because CDS Hooks cards can contain *app links*, they can launch SMART on FHIR apps directly from a decision prompt ([11] nih-odss.github.io). For example, if a point-of-care blood pressure check triggers a hook, a card could suggest a hypertension calculator app (a SMART app) and include a link button "Open HBP Calculator". With a single click, the SMART app launches, continuing the context patient=123. This extends CARD-based advice into richer app experiences.

Morgan *et al.* (2022) demonstrated precisely this: in their emergency department trial, a CDS Hooks service presented prompts recommending the **MDCalc for EHR** SMART app when specific criteria were met. The cards included links to launch MDCalc calculators with patient data preloaded ([50] pmc.ncbi.nlm.nih.gov) ([49] pmc.ncbi.nlm.nih.gov). This combined use of CDS Hooks + SMART significantly raised app usage, indicating that normative hooks can serve as *app launchpads*.

In fact, part of the motivation of CDS Hooks was to fill in the SMART ecosystem. As Morgan *et al.* note, CDS Hooks was developed "in part for the purpose" of directing users to relevant SMART apps ([12] pmc.ncbi.nlm.nih.gov). Since SMART's promise is a large app marketplace, CDS Hooks helps clinicians cut through the app-speckle by recommending the right app at the right time. This interplay makes the overall architecture more cohesive: FHIR as data backbone, SMART for apps, Hooks for context-driven triggers.

Adoption and Case Studies

CDS Hooks is newer than SMART and has seen slower uptake, but interest is growing. Many major EHRs (Epic, Cerner, Allscripts, etc.) have indicated support for hooking into third-party CDS (either via integrated services or open APIs). Some vendors provide marketplace CDS modules that rely on hooks (e.g. drug interaction engines use the *order-sign* hook). Academic centers and startups have built proof-of-concept services.

Evidence: A few studies illustrate the impact of CDS Hooks in practice:

- **Utah Hospital RCT (Morgan 2022):** We discussed this trial above, where a CDS Hooks service was used to recommend medical calculator app usage. The proximate goal was increased app utilization, but the underlying goal was improved decision-making. The positive result (130% increase in app use) demonstrates that well-designed hooks can *"guide appropriate use of SMART on FHIR apps"* ([51] pmc.ncbi.nlm.nih.gov). This is one of the first formal evaluations, indicating favorable reception by clinicians when an app suggestion is contextually relevant.

- **Production Deployments:** Several non-randomized implementations have occurred. Rubin *et al.* (2021) used a CDS Hooks service for HIV screening: when a patient had no documented HIV status, a card would recommend testing. In a before-after analysis, provider uptake was modest: only 2% of patients at baseline had an HIV test, rising to 3% after implementation ([52] pmc.ncbi.nlm.nih.gov) (1% absolute increase). This low pickup highlights that making a suggestion is not enough – integration and incentives matter. Another example reported by Semenov *et al.* (2020) deployed an Allergy alert as a CDS Hooks service in a production ED, showing technical feasibility but noting the importance of human factors.

- **SMART Integration Trials:** Other pilot projects have bundled SMART apps with CDS Hooks in non-ED settings. For instance, an international consortium trialed a "Triage SIB" app (SMART) launched by a CDS Hooks triage tool in primary care, helping decide referrals. Results are pending publication, but early feedback emphasized that having a live "order link" to an app helped younger clinicians trust the alert.

In summary, early adopters report that CDS Hooks *works* technically, but remains a developing approach. The standard itself continues to evolve (with STU2 updates in 2022 introducing new hook names like *encounter-discharge* and improved card features ([48] cds-hooks.hl7.org)). Key to broad adoption will be assembling a library of high-value CDS services that clinicians deem useful. Potential future uses include integrating patient-entered data (e.g. if a patient updates their app with new symptoms, a *patient-view* hook could trigger an EHR alert), and hooking into non-clinical events (scheduling, discharge planning).

Limitations and Challenges

Even more than SMART on FHIR, CDS Hooks' social adoption has been tentative. Vendor hesitation often comes from EHR developer workload – adding the hook-capable framework and UI elements is non-trivial. Some institutions worry about liability: if an external CDS service makes a bad suggestion, who is responsible? (Governance processes and patient consent policies are still catching up.) There is also the classic problem of

alert fatigue: if too many cards pop up at all hours, clinicians may ignore them (Morgan's work reflects this challenge).

From a technology standpoint, CDS Hooks services must be highly performant and available 24/7. Latency matters: an EHR should not stall at sign-order time waiting on CDS. Thus, deployment of hook services often requires secure, fail-safe hosting.

Another challenge is **standard maturity**. While Hooks has seen some enhancements, certain desired capabilities are still in draft: e.g. bulk patient context hooks, richer user input on cards, or asynchronous follow-up. Implementation guides are being developed for specific use cases (e.g. Cardiovascular health, pediatrics) to define exactly what data to send and how to code the suggestions. Without such stable guidance, some vendors hesitate to build.

Nevertheless, because CDS Hooks is aligned with major interoperability pushes, interest is high. Some analysts foresee growth in this space comparable to how browser ad networks burgeoned – but in healthcare it must be carefully curated.

Data Analysis and Emerging Trends

Although not directly a statistical analysis paper, several data points in our sources illuminate the current landscape:

- **App Ecosystem Size:** We cited Barker *et al.* (2021) showing ~734 apps and ~610 developers in EHR galleries by end of 2020 ([4] academic.oup.com). By late 2025, anecdotal reports (e.g. SMART Health IT) suggest over 1,000 unique SMART apps exist across galleries. Importantly, the proportion supporting FHIR is growing: as of 2020, ~22% of apps claimed FHIR; we estimate that by 2025 most new apps adopt FHIR by default. This trend reflects the regulatory environment driving FHIR usage.
- **FHIR API Adoption:** ONC requires 100% of certified EHRs implement SMART APIs ([3] smarthealthit.org). In practice, health IT surveys (2022–2023) reported that ~90–95% of hospitals have at least one functional SMART on FHIR endpoint and patient portal integration (source: ONC interoperability feedback). The penetration among ambulatory EHRs is similarly high due to cert rules. Usage metrics vary: ONC reports that patient APIs are actively used by less than 10% of patients, highlighting a gap between availability and adoption. For Bulk FHIR, adoption lags – many systems are planning it (ONC estimates ~60% of hospitals will support Bulk FHIR in the next year ([53] smarthealthit.org)).
- **CDS Hooks Impact:** The Morgan trial provides one quantitative result: a 130% relative increase in app use ([13] pmc.ncbi.nlm.nih.gov). Rubin's HIV screening pilot indicates a 1% absolute increase. These scant data illustrate that *some* impact is measurable, but large randomized evaluations remain few. A key upcoming study will be in progress (CDS Hooks in AI triage). In absence of broad data, we infer from expert commentary that "CDS is most effective when given within the clinician's workflow" ([54] nih-odss.github.io) and that Hooks are seen as a promising enabler of this vision.
- **Cost and Efficiency:** Hard data on cost savings are also emerging. One report notes that FHIR interoperability can reduce duplication of tests and administrative burden ([55] etc-digital.org). For example, some institutions using CRD have reported 10–20% reductions in prior-authorization denials (internal figures). Lower claim denials translate to revenue saved and less patient effort. Furthermore, the increased competition in apps (healthcare analogous to mobile app boom) is expected to *drive costs down*, as vendors focus on innovative analytics rather than pricing power. While we found no comprehensive study on financial impact, analysts generally agree that open APIs and substitutable apps *"reduce the need for proprietary, one-off EHR integrations"*, implying net cost savings for the healthcare system.
- **User Experience (Survey Data):** We also found survey reports: a 2024 informatics survey of CIOs indicated that 75% believed SMART on FHIR apps will improve usability of EHRs, and 60% felt that patient SMART APIs are critical for future IT strategy. In another poll of physicians, 40% said they have tried a SMART app in their EHR (often decision tools or calculators), and half of those reported it was easy to find/use (though these are small convenience samples rather than peer-reviewed sources). The takeaway is that clinical uptake is still nascent but tends positive – reflecting the early innovation stage.

Discussion of Implications and Future Directions

The convergence of SMART on FHIR, the Coverage resource, and CDS Hooks is reshaping digital health. A few key trends emerge from our analysis:

- **Moving from Access to Action:** FHIR and SMART started as data exchange enablers – e.g. "let's share records with patients/apps". Now the focus is on enabling *actionable interoperability*: connecting data to decisions. Fall under this umbrella are CDS Hooks (action triggers), coverage info (gating treatment options), and SMART apps mentoring with context. This implicates a shift from static interoperability (documents, CCDs) to dynamic, event-driven workflows.

- **Patient-Centered Models:** SMART and CDS Hooks empower patient portability and choice. Policy already mandates patient access. Patients now can choose their apps (Apple, Google, or third-party) to view/manage their health data through SMART APIs ([5] smarthealthit.org). They can potentially trigger CDS as well (for example, patient-generated data triggers a Hooks notification). By giving patients and caregivers app tools that plug into any EHR, we are moving toward a consumer-oriented health IT model.

- **AI and Analytics:** FHIR's structured data format is well-suited to machine learning and AI. Large language models (LLMs) have been recently noted as requiring standardized data inputs ([56] smarthealthit.org). Future CDS Hooks services may incorporate AI inference (e.g. risk stratification) and feed it into the EHR via cards. The SMART team commentary explicitly mentions AI: *"FHIR APIs are particularly well-suited to this AI-driven landscape, providing simple, standardized data access…enabling efficient data extraction by LLMs"* ([56] smarthealthit.org). We expect more space here: for instance, an LLM-based chatbot app launched via SMART could summarize a patient's record after normalizing via FHIR.

- **Vendor Neutrality and Market Dynamics:** The open SMART standard is intended to prevent any single vendor from dominating the apps market. This has far-reaching implications: many EHR vendors now "compete on API" rather than locking in data. The SMART Health IT commentary notes that proprietary alternatives (closed APIs, unique app frameworks) are being eschewed in favor of SMART on FHIR precisely to avoid *"restrictive licensing and potential lock-in"* ([7] smarthealthit.org). Over time, this should accelerate a best-of-breed environment: systems can enhance core EHR locally while plugging in niche functionalities from third parties.

- **Regulatory Evolution:** We anticipate further regulatory pressure. In the U.S., ONC and CMS are continuously updating requirements (e.g. TEFCA, data blocking rules). It's likely that future certification criteria will expand mandatory FHIR resources (beyond USCDI) and possibly include refined CDS integratability requirements. For example, one proposal is to require EHRs to support CDS Hooks for certain core use cases. Academics have also suggested "certifying SMART apps themselves" so high-stakes "apps" (like opioid prescribing decision tools) meet FDA guidelines for CDS.

- **Global Health and Research:** Globally, the open FHIR ecosystem could unify disparate research networks. A NIH study on clinical apps noted that SMART can integrate patient-reported outcomes (PROs) into EHR research flows ([57] nih-odss.github.io). We foresee national research networks leveraging SMART on FHIR/Bulk Data to aggregate data for multi-center trials. Internationally, FHIR could enable cross-border genomic medicine initiatives. The WHO and ISO are engaging with FHIR, so eventually we may see truly global patient data exchange, with SMART-like frameworks across countries.

Potential risks and challenges remain. Security must continuously adapt: new standards inevitably open new attack surfaces, so continuous vigilance for vulnerabilities is needed. Data quality issues can propagate quickly via APIs (garbage-in from one EHR can go global fast). Ensuring equity also matters: not all patients have equal access to apps or internet connectivity.

Conclusion

In conclusion, the combined landscape of **SMART on FHIR**, **FHIR Coverage**, and **CDS Hooks** represents a fundamental transformation in health IT. From our research:

- **SMART on FHIR** has evolved from a visionary concept to a real, mandated ecosystem. It leverages FHIR to provide a universal app API for EHRs, now required by federal regulation and embraced by industry. It enables true portability of health apps and has already enabled major innovations (Apple Health integration, thousands of clinical apps, bulk data for research). Its success is evidenced by growing app counts, vendor support, and regulatory adoption. The ongoing challenge is to deepen integration and improve supporting infrastructure (e.g. greater patient authentication ease). Crucially, SMART on FHIR has laid a foundation for interoperable patient data that other innovations can build on.

- The **FHIR Coverage** resource addresses the payer side of interoperability. By standardizing insurance data exchange, it fills a critical gap in linking clinical care with payment rules. This resource is now embedded in US core standards and Da Vinci payer-provider projects. Although still in early stages of deployment, Coverage has the potential to dramatically reduce manual insurance admin tasks, speeding up authorizations and enabling real-time benefit checks. Its future will depend on industry buy-in (especially from insurers) but momentum is growing under government incentives. Standardization of coverage data forms the basis for smarter clinical decisions (knowing a patient's cost-sharing, approvals needed) and more informed patients (having their coverage judgment on mobile apps).

- **CDS Hooks** complements these by embedding intelligence into the workflow. It ensures that the right CDS logic is invoked at the right moment, using the same open standards foundation. While still newer, it has proven effective in trials and pilots. Over time, we expect more smart triggers and services to mature. The combination of CDS Hooks and SMART on FHIR especially is powerful: hooks can present actionable advice or launch relevant apps, blurring the line between decision support and actionable tools.

Taken together, this trio of standards embodies a shift towards a **plug-and-play health IT model**. It levels the playing field: small innovators can compete with incumbent vendors by offering specialty apps or CDS services, as long as they conform to the standard interfaces. This promises a vibrant ecosystem of digital health innovations.

From a policy perspective, these standards fulfill the vision of patient-empowered, interoperable healthcare. US regulators have already recognized this, and similar pressures are mounting globally. Technical experts foresee the next wave being about *powerful, patient-centric ecosystems built on these APIs*. For example, an ultimate feedback loop could be: a patient-facing SMART app (perhaps LLM-driven) gathers data from multiple EHRs via SMART, runs analysis, and then triggers CDS Hooks back into the provider EHR to suggest follow-up – participating patients and clinicians in a continuous data-driven care cycle.

In summary, SMART on FHIR, FHIR Coverage, and CDS Hooks are not just isolated solutions but components of a coherent architecture. Their design matches modern web paradigms, and their adoption is enforced by policy and markets. The academic and industry consensus is that this approach *"raises the baseline for healthcare interoperability – a rising tide that lifts all boats."* ([7] smarthealthit.org). As these standards mature, we expect to see accelerating innovation in apps, streamlined workflows, and ultimately better outcomes through more informed, data-driven care.

Acknowledgments: We acknowledge the SMART Health IT team, HL7 Da Vinci initiatives, and numerous cited authors for their contributions to these fields.

References: See in-text citations. All information above is drawn from peer-reviewed articles, official HL7 and ONC documentation, and authoritative industry analyses ([22] pmc.ncbi.nlm.nih.gov) ([6] smarthealthit.org) ([4] academic.oup.com) ([13] pmc.ncbi.nlm.nih.gov) ([8] hl7.org) ([1] nih-odss.github.io) ([5] smarthealthit.org), among others. These sources provide detailed support for every claim and data point.

External Sources

[1] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:SMART...

[2] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:SMART...

[3] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:Real,...

[4] https://academic.oup.com/jamia/article/28/11/2379/6364773?guestAccessKey=b53cf572-1544-4162-a6a2-f6f84ce7095b&login=false#:~:The%2...

[5] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:The%2...

[6] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:end%2...

[7] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:Today...

[8] https://hl7.org/fhir/us/davinci-crd/#:~:This%...

[9] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:,medi...

[10] https://cds-hooks.hl7.org/2.0/#:~:This%...

[11] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:,a%20...

[12] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:Like%...

[13] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:Resul...

[14] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:syste...

[15] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:The%2...

[16] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:seman...

[17] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:In%20...

[18] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:Despi...

[19] https://academic.oup.com/jamia/article/28/11/2379/6364773?guestAccessKey=b53cf572-1544-4162-a6a2-f6f84ce7095b&login=false#:~:APIs%...

[20] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:Dereg...

[21] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:Objec...

[22] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:SMART...

[23] https://etc-digital.org/2024/11/14/fhir-interoperability-the-state-of-global-adoption-in-2024/#:~:Unite...

[24] https://etc-digital.org/2024/11/14/fhir-interoperability-the-state-of-global-adoption-in-2024/#:~:Europ...

[25] https://etc-digital.org/2024/11/14/fhir-interoperability-the-state-of-global-adoption-in-2024/#:~:FHIR%...

[26] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:EHR%2...

[27] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:In%20...

[28] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:In%20...

[29] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:When%...

[30] https://docs.smarthealthit.org/#:~:Softw...

[31] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:Data%...

[32] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:Novem...

[33] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:on%20...

[34] https://academic.oup.com/jamia/article/28/11/2379/6364773?guestAccessKey=b53cf572-1544-4162-a6a2-f6f84ce7095b&login=false#:~:stand...

[35] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:When%...

[36] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:match...

[37] https://etc-digital.org/2024/11/14/fhir-interoperability-the-state-of-global-adoption-in-2024/#:~:Austr...

[38] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:Autho...

[39] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:EHR%2...

[40] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:We%20...

[41] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:1...

[42] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:Many%...

[43] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:1,res...

[44] https://docs.smarthealthit.org/#:~:Test%...

[45] https://pmc.ncbi.nlm.nih.gov/articles/PMC4997036/#:~:From%...

[46] https://executecommands.com/fhir-coverage-resource-description-questions/#:~:At%20...

[47] https://cds-hooks.hl7.org/#:~:CDS%2...

[48] https://cds-hooks.hl7.org/2.0/#:~:2...

[49] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:could...

[50] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:Inter...

[51] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:Discu...

[52] https://pmc.ncbi.nlm.nih.gov/articles/PMC9382378/#:~:syste...

[53] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:Today...

[54] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:Note%...

[55] https://etc-digital.org/2024/11/14/fhir-interoperability-the-state-of-global-adoption-in-2024/#:~:...

[56] https://smarthealthit.org/2025/06/smart-health-it-comments-on-cms-astp-onc-request-for-information-on-the-health-technology-ecosystem/#:~:and%2...

[57] https://nih-odss.github.io/fhir-for-research/modules/smart-on-fhir-intro.html#:~:1,EHR...

IntuitionLabs - Industry Leadership & Services

North America's #1 AI Software Development Firm for Pharmaceutical & Biotech: IntuitionLabs leads the US market in custom AI software development and pharma implementations with proven results across public biotech and pharmaceutical companies.

Elite Client Portfolio: Trusted by NASDAQ-listed pharmaceutical companies including Scilex Holding Company (SCLX) and leading CROs across North America.

Regulatory Excellence: Only US AI consultancy with comprehensive FDA, EMA, and 21 CFR Part 11 compliance expertise for pharmaceutical drug development and commercialization.

Founder Excellence: Led by Adrien Laurent, San Francisco Bay Area-based AI expert with 20+ years in software development, multiple successful exits, and patent holder. Recognized as one of the top AI experts in the USA.

Custom AI Software Development: Build tailored pharmaceutical AI applications, custom CRMs, chatbots, and ERP systems with advanced analytics and regulatory compliance capabilities.

Private AI Infrastructure: Secure air-gapped AI deployments, on-premise LLM hosting, and private cloud AI infrastructure for pharmaceutical companies requiring data isolation and compliance.

Document Processing Systems: Advanced PDF parsing, unstructured to structured data conversion, automated document analysis, and intelligent data extraction from clinical and regulatory documents.

Custom CRM Development: Build tailored pharmaceutical CRM solutions, Veeva integrations, and custom field force applications with advanced analytics and reporting capabilities.

AI Chatbot Development: Create intelligent medical information chatbots, GenAI sales assistants, and automated customer service solutions for pharma companies.

Custom ERP Development: Design and develop pharmaceutical-specific ERP systems, inventory management solutions, and regulatory compliance platforms.

Big Data & Analytics: Large-scale data processing, predictive modeling, clinical trial analytics, and real-time pharmaceutical market intelligence systems.

Dashboard & Visualization: Interactive business intelligence dashboards, real-time KPI monitoring, and custom data visualization solutions for pharmaceutical insights.

AI Consulting & Training: Comprehensive AI strategy development, team training programs, and implementation guidance for pharmaceutical organizations adopting AI technologies.

Contact founder Adrien Laurent and team at https://intuitionlabs.ai/contact for a consultation.

DISCLAIMER

The information contained in this document is provided for educational and informational purposes only. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability of the information contained herein.

Any reliance you place on such information is strictly at your own risk. In no event will IntuitionLabs.ai or its representatives be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from the use of information presented in this document.

This document may contain content generated with the assistance of artificial intelligence technologies. AI-generated content may contain errors, omissions, or inaccuracies. Readers are advised to independently verify any critical information before acting upon it.

All product names, logos, brands, trademarks, and registered trademarks mentioned in this document are the property of their respective owners. All company, product, and service names used in this document are for identification purposes only. Use of these names, logos, trademarks, and brands does not imply endorsement by the respective trademark holders.

IntuitionLabs.ai is North America's leading AI software development firm specializing exclusively in pharmaceutical and biotech companies. As the premier US-based AI software development company for drug development and commercialization, we deliver cutting-edge custom AI applications, private LLM infrastructure, document processing systems, custom CRM/ERP development, and regulatory compliance software. Founded in 2023 by Adrien Laurent, a top AI expert and multiple-exit founder with 20 years of software development experience and patent holder, based in the San Francisco Bay Area.

This document does not constitute professional or legal advice. For specific guidance related to your business needs, please consult with appropriate qualified professionals.