

# **Getting Started with open.epic**

# Presenters



**Josh  
Priebe**

Integrations Technical Services



**Sean  
Hubber**

Software Developer



# Learning Objectives

1. Understand Epic's standards-first approach to data exchange
2. Walk the road(map) from app concept to Go-Live
3. Strategies for successful connectivity and collaboration with customers

## Epic's Developer Guide

Depending on the technologies used, most applications will require a client ID that is issued from Epic. Developers may obtain client IDs from [open.epic.com](https://open.epic.com). Customers may then request that the client ID be synced to their instance, and app developers may directly approve these, without involvement from Epic. Here's a guide to get you started:



# Agenda

## **01 Data Sharing Philosophy and Design**

Overview of supported standards, our websites, and architecting your data exchange

## **02 Register a Client Record**

Obtain client IDs for implementation of OAuth 2.0

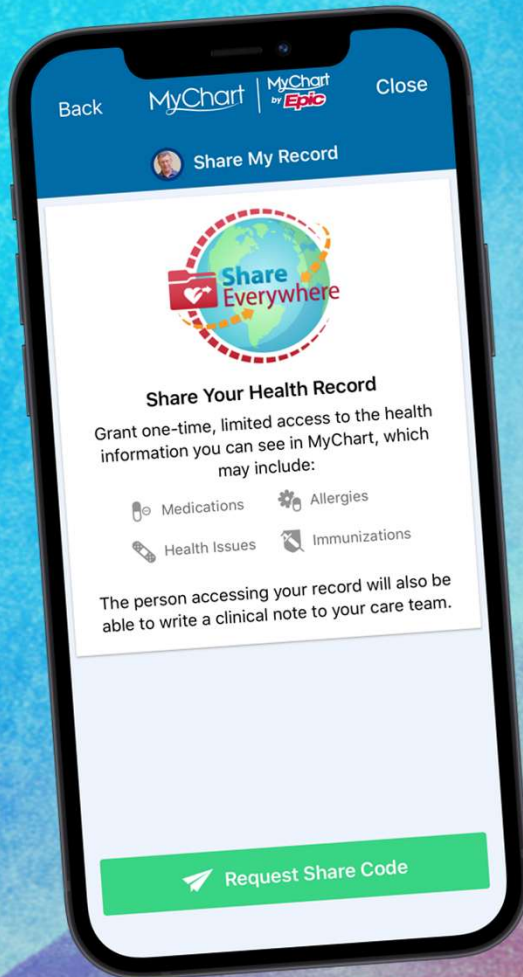
## **03 Develop and Test**

Simulate app launches and connectivity by connecting to our FHIR developer sandbox

## **04 Customer Implementation and Going Live**

Strategize your install project and Go Live with Epic customers

# Sharing Data *with* Patients & Providers



**14 million** patient charts exchanged daily  
**>52%** of exchanges are with non-Epic systems



1

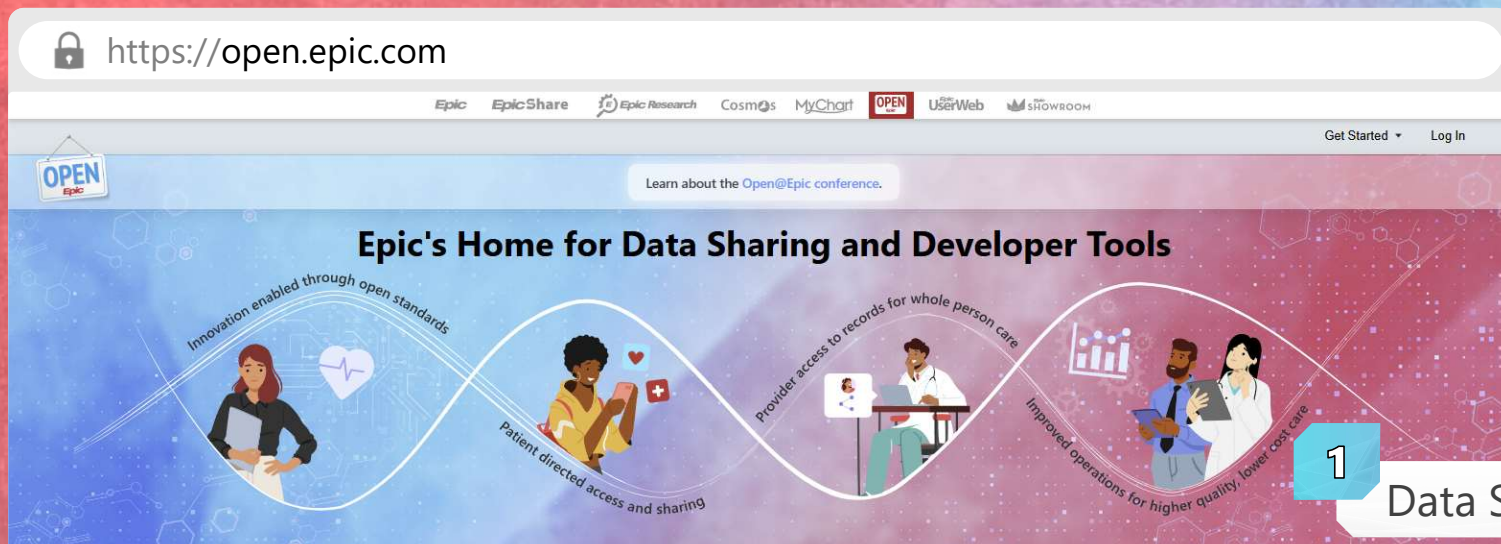
Data Sharing Design



# Generalize *to* Benefit More *with* Industry Standards



**500+** FHIR APIs



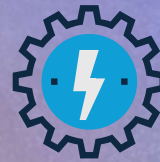
1

Data Sharing Design

# Interfaces



**+368 billion**  
*interface messages  
annually*



**+55,000**  
*active interfaces*



**+2,600**  
*vendors*

1

Data Sharing Design



# open.epic.com



## Paths Epic Provides *for* Third Parties *to* Connect *available on open.epic.com*



**500+ FHIR APIs**  
*for USCDI requirements  
and much more*



**180+ standards-  
based interfaces**  
*including HL7, DICOM,  
ASC X12, and more*





**150+ Epic-created  
public APIs**  
*for scenarios where  
there was no standard*

1

Data Sharing Design



# Data Exchange Tutorial




## Design Overview

Epic supports a wide variety of data exchange technologies. This guide provides an overview of design considerations and the different ways to plan your connection to organizations using Epic.

### What Should I Consider When Designing My Data Exchange?

Due to the vast breadth of standards and types of APIs and interfaces a developer may wish to use, there are many factors to consider when determining the best way for you to exchange data with Epic. The sections below cover the most common considerations when designing a connection, while the [Case Studies](#) section toward the end of this document provides examples for applying these considerations to some sample scenarios.



#### TYPES OF DATA EXCHANGED


FINANCIAL

PROVIDER

CLINICAL

REGISTRATION

SCHEDULING




#### DIRECTION OF DATA EXCHANGE

PUSHED INTO EPIC

PUSHED OUT OF EPIC

PULLED INTO EPIC

PULLED OUT OF EPIC



#### WORKFLOWS USING DATA EXCHANGE

FINANCIAL


REPORTING

CLINICAL

REGISTRATION

SCHEDULING

BACKEND PROCESS



#### DATA EXCHANGE METHOD

HL7V2

HL7V3

FHIR

ASC X12


EPIC PROPRIETARY API

IHE

KIT

CONTENT

DICOM



#### SECURITY MECHANISMS

TCP/IP

HTTP GET

OAUTH 2.0

SAML

BASIC AUTHENTICATION

### Types of Data to Exchange

Before you begin designing your connection, you must first understand what data you plan to exchange with Epic. Start by defining the discrete types of data your app will work with, and define a scope for the level of detail required within that data set. For example, you may need to exchange medication information. You should then more specifically define data elements you need within that data type, such as dosage information, RxNorm codes, etc.

# Data Exchange Tutorial



Standards-Based  
Interfaces



RESTful FHIR APIs



Public APIs



CDS Hooks



Automated CDA



Bulk FHIR



SMART on FHIR



Bulk FHIR



CDS Hooks

1

Data Sharing Design

*When designing a new integration, start with the use case.  
The technology will follow.*



# Common Interfaces & Use Cases



Appointment Scheduling  
Blood Transfusions  
Financial Transactions  
Flowsheet and Device Data  
Inventory/Supply Management  
Medication Administrations

Orders and Results  
Patient Administration  
Referral Management  
Surgical Scheduling/Tracking  
Transcriptions/Documents  
Vaccine Administrations

# FHIR APIs & Use Cases



**138 billion messages**  
in the past year

10.1.3 Resource Content

Structure UML XML JSON Turtle R3 Diff All

Structure

Name	Flags	Card.	Type	Description & Constraints
Observation	I[N]		DomainResource	Measurements and simple assertions + Rule: dataAbsentReason SHALL only be present if Observation.value[x] is not present + Rule: If Observation.code is the same as an Observation.component.code then the value element associated with the code SHALL NOT be present Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension
identifier	Σ	0..*	Identifier	Business Identifier for observation
basedOn	Σ	0..*	Reference(CarePlan   DeviceRequest   ImmunizationRecommendation   MedicationRequest   NutritionOrder   ServiceRequest)	Fulfills plan, proposal or order
partOf	Σ	0..*	Reference(MedicationAdministration   MedicationDispense   MedicationStatement   Procedure   Immunization   ImagingStudy)	Part of referenced event
status	?! Σ	1..1	code	registered   preliminary   final   amended + ObservationStatus (Required)
category		0..*	CodeableConcept	Classification of type of observation Observation Category Codes (Preferred)
code	Σ	1..1	CodeableConcept	Type of observation (code / type) LOINC Codes (Example)
subject	Σ	0..1	Reference(Patient   Group   Device   Location)	Who and/or what the observation is about
focus	Σ[TU]	0..*	Reference(Any)	What the observation is about, when it is not about the subject of record
encounter	Σ	0..1	Reference(Encounter)	Healthcare event during which this observation is made
effective[x]	Σ	0..1		Clinically relevant time/time-period for observation



# Data Sharing Playbooks

 <https://open.epic.com/Playbooks>

Epic EpicShare Epic Research CosmOS MyChart OPEN UserWeb sTOWROOM

Get Started Log In



## Data Sharing Playbooks

Data Sharing Playbooks offer practical recommendations for connecting with Epic community members. Each playbook highlights proven approaches to common data sharing use cases, whether supporting patient-directed access, enabling providers to deliver whole-person care, or improving operations. By leveraging open standards, these playbooks help innovators accelerate safe, reliable connections.



All Playbooks



Patient Access



Provider Workflows



Operational Efficiency

### Alert Managers

Alert Managers ingest alerts from a variety of devices, such as inpatient vitals monitors, and determine who should get notified for each alert via Epic's Alert Communicator.

### Ambulatory Cardiac Devices

Wearable cardiac devices like Holter monitors, event monitors, long-term continuous monitors and mobile cardiac telemetry, are used to monitor and record cardiac data while a patient is away from the office or hospital.

### Alert Managers

Alert Managers ingest alerts from a variety of devices, such as inpatient vitals monitors, and determine who should get notified for each alert via Epic's Alert Communicator.

### Ambulatory Cardiac Devices

Wearable cardiac devices like Holter monitors, event monitors, long-term continuous monitors and mobile cardiac telemetry, are used to monitor and record cardiac data while a patient is away from the office or hospital.

### Automated Dispensing System

Automated Dispensing Systems (ADS) are computerized devices that store, dispense, and track medications at the point of service by receiving patient and order information to determine available medications and send dispense information back to Epic when medications are removed.

### Bayesian Medication Dosing Decision Support

1

Data Sharing Design

# Develop & Test





# Building FHIRAPI Calls / Structure

Observation . Read (Vital Signs) (R4)

“Resource” “Action” “Subresource” “Version”

# Building FHIR API Calls / Documentation

[Try It](#)

## Observation.Read (Vital Signs) (R4) USCDI

**General Information**

HTTP Method:	GET
URL Template:	api/FHIR/R4/Observation/{ID}
Supported OAuth 2.0 User Types:	Backend Systems and Non-OAuth 2.0, Clinicians or Administrative Users, Patients

**Description**

The FHIR Observation resource defines measurements and assertions about a patient, including vital signs, laboratory data, imaging results, devices measurements, clinical assessment tools, personal characteristics, social history, and core characteristics.

The following types of vital information and corresponding LOINC codes are supported by default. How specific types of data translate from within the Epic system to FHIR might vary depending on organizational configuration:

- 2708-6 Oxygen saturation in arterial blood
- 2710-2 Oxygen saturation in capillary blood by oximetry
- 29463-7 Body weight
- 3141-9 Body weight measured
- 3150-0 Inhaled oxygen concentration
- 3151-8 Inhaled oxygen flow rate
- 59408-5 Oxygen saturation in arterial blood by pulse oximetry (SpO2)
- 8287-5 Head occipital-frontal circumference by tape measure
- 8302-2 Body height



# Building FHIR API Calls / Documentation

## Native Request Elements

Name	Description	Is Optional	Is Array
<b>category</b> (String)	Use "vital-signs" to search for vitals observations.	conditional ⓘ	false
<b>code</b> (String)	LOINC code, CADSR code, flowsheet ID, or encoded flowsheet ID. Either this element or category must be provided.  The code element value varies depending upon what is passed (for example, a flowsheet ID vs. a LOINC code). If using a flowsheet ID, the system value is different between Epic organizations, and it is also different between production and non-production environments for the same organization.	conditional ⓘ	false
<b>date</b> (String)	The date range for when the observation was taken.  For growth chart data (Epic version August 2021 and later), only the most recent observation within the timeframe is returned.	true	false
<b>patient</b> (String)	Reference to a patient resource the observation is about. Either this element or subject must be provided. If both are provided, they must match.	conditional ⓘ	false
<b>subject</b> (String)	Reference to a patient resource the observation is about. Either this element or patient must be provided. If both are provided, they must match.	conditional ⓘ	false

3

Develop & Test

# Building FHIR API Calls / Documentation

## Post-filter Request Elements

Starting in the May 2024 version of Epic, the following search parameters that use a post-filtering mechanism are available. When responding to a request, the Epic FHIR server first retrieves all results that match your search (using any native search parameters you've provided), then filters down those results based on the additional post-filtered parameters you've specified.

For more information about post-filter parameters and related considerations, refer to the General Considerations section of the [FHIR Search Parameters](#) document.


Name	Description	Is Optional	Is Array
<b>based-on</b> (String)	Unsupported	true	false
<b>combo-code</b> (String)	Matches to the code or component.code element in the response.	true	false
<b>combo-data-absent-reason</b> (String)	The reason why the expected value in the element Observation.value[x] or Observation.component.value[x] is missing.	true	false
<b>combo-value-concept</b> (String)	The value or component value of the observation, if that value is a CodeableConcept. Matches to valueCodeableConcept or component.valueCodeableConcept.	true	false
<b>component-code</b> (String)	Matches to the component.code element in the response.	true	false
<b>component-data-absent-reason</b> (String)	The reason why the expected value in the element Observation.component.value[x] is missing.	true	false
<b>component-value-concept</b> (String)	The component value of the observation, if that value is a CodeableConcept. Matches to component.valueCodeableConcept.	true	false
<b>data-absent-reason</b> (String)	The reason why the expected value in the element Observation.value[x] is missing.	true	false
<b>derived-from</b> (String)	Unsupported	true	false
<b>device</b> (String)	Unsupported	true	false
<b>encounter</b> (String)	Encounter associated with this observation value, if applicable. When multiple encounters are involved, such as for growth	true	false

3

Develop & Test



# Building FHIR API Calls / Documentation




## Observation.Read (Vital Signs) (R4) USCDI

### General Information

HTTP Method:

URL Template:

Supported OAuth 2.0 User Types: 

GET

api/FHIR/R4/Observation/{ID}

Backend Systems and Non-OAuth 2.0, Clinicians or Administrative Users, Patients

### Description

The FHIR Observation resource defines measurements and assertions about a patient, including vital signs, laboratory data, imaging results, devices measurements, clinical assessment tools, personal characteristics, social history, and core characteristics.

The following types of vital information and corresponding LOINC codes are supported by default. How specific types of data translate from within the Epic system to FHIR might vary depending on organizational configuration:

- 2708-6 Oxygen saturation in arterial blood
- 2710-2 Oxygen saturation in capillary blood by oximetry
- 29463-7 Body weight
- 3141-9 Body weight measured
- 3150-0 Inhaled oxygen concentration
- 3151-8 Inhaled oxygen flow rate
- 59408-5 Oxygen saturation in arterial blood by pulse oximetry (SpO2)
- 8287-5 Head occipital-frontal circumference by tape measure
- 8302-2 Body height

# Building FHIR API Calls / Testing & Try-It Cases

**Observation.Read (Vital Signs) (R4)** USCDI

Http Method  
GET

URL Template  
`https://fhir.epic.com/interconnect-fhir-oauth/api/FHIR/R4/Observation/{ID}`

Request Parameters  
**ID**  
envjcVAhuFtXhXNFlg1Dr-2-8diVcq3BOMcZpbjYOC7JAJ1pPzK0v1075T4XMHL83

+ Raw Request

Try It Out!

Response

resourceType	Observation
id	envjcVAhuFtXhXNFlg1Dr-2-8diVcq3BOMcZpbjYOC7JAJ1pPzK0v1075T4XMHL83
status	final
category	
coding	
system	http://terminology.hl7.org/CodeSystem/observation-category
code	vital-signs
display	Vital Signs
text	Vital Signs
code	
coding	
system	urn:oid:1.2.840.114350.1.13.0.1.7.2.707679
code	5
display	BP
system	http://open.epic.com/FHIR/StructureDefinition/observation-flowsheet-id
code	tBdNYepLeojPG60x7nUx9kQ0
display	BP
system	urn:oid:1.2.246.537.6.96

3

Develop & Test



**Demo: Try It!**

3

Develop & Test

# OAuth2 & Client Records



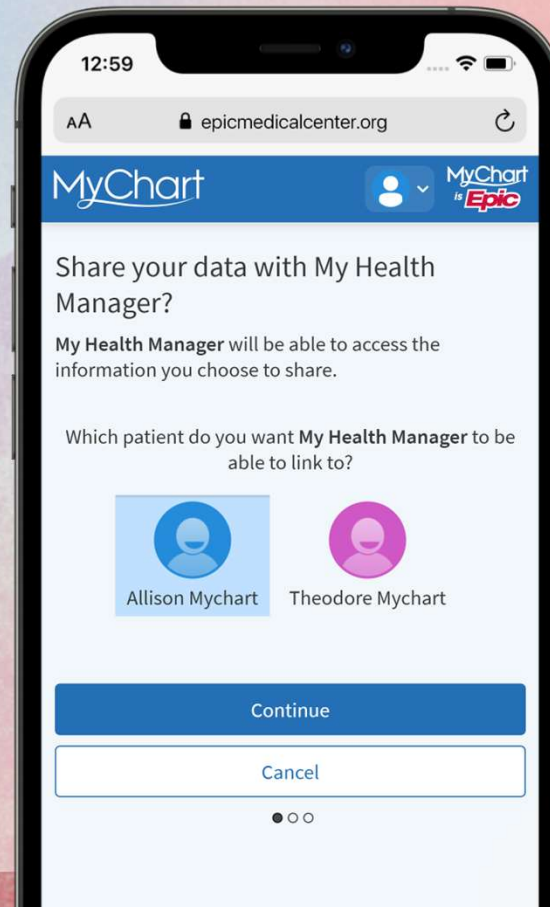


# OAuth 2.0 in a Nutshell

Health app, what  
meds do I need to  
take today?



Can I get an access  
token to check?



3

Develop & Test

# App Contexts

Patient



Clinician/Staff



Backend System



2

Register a Client Record



# Demo: App Registration!

3

Develop & Test

# Deploying and Collaborating





# Collaboration & Project Planning


## Install Tips

- Involve customer operations and IT teams early in the project
  - Customer staff can contact their Epic representatives for expert help
- Each customer has their own instance of Epic
  - Expect variation in mappable data elements, workflows, patient identity, etc.
- Reference our technology-specific app implementation briefs



# Collaboration & Project Planning

- Reference our technology-specific app implementation briefs

 <https://fhir.epic.com/Documentation?docId=implementing>

## Implementing a SMART on FHIR EHR or Standalone Launch

Many apps launch from a user workflow in Hyperspace to an external web application and use a single-sign-on workflow through SMART on FHIR to log the user in to the external page. SMART on FHIR is the recommended practice for app integrations that launch from Epic. For more information on building with this technology, see our [SMART on FHIR launch simulator](#), and our [Hyperdrive Test Harness](#) for testing out your integration using self-service tools.

SMART on FHIR is unique in its support of standalone launch. During a standalone launch, an app can redirect the Epic user to an Epic login page. By authenticating, the user authorizes the app to access information from Epic. You will need to verify that your app has the correct user type set and understand which login credentials can be used in each case. This is related to the "Who will primarily be using this app?" question. If you have selected:

- Patients: Users must use their MyChart login credentials to authenticate.
- Clinicians or Administrative Users: Users must use their EMP login credentials to authenticate.

As you develop your SMART on FHIR integration, consider how to make your app's integration as performant as possible by limiting FHIR API calls or performing them asynchronously from the web page load. Users want to interact with your app right away, not wait for it to load.

## Information to send to the Customer

When you've tested it out and are ready to implement your SMART on FHIR app with a customer, the customer will need just a couple of pieces of technical information to configure the SMART on FHIR launch in Epic. This is the same information you would have used yourself in the SMART on FHIR simulator:

1. **Client IDs** – the organization will follow the [App Request](#) process to download your client ID to prepare for your install. For an EHR launch, they will use either the non-production or production client ID in their SMART on FHIR launch configuration, depending on the environment.
2. **Launch URL** – For an EHR launch, the organization's application build team will need the launch URL for the initial landing page that kicks off your SMART on FHIR launch's OAuth 2.0 handshake.
3. **Tokens in OAuth 2.0 Context** – for an EHR launch, the organization's application build team will need the list of context tokens that your app needs at the point of launch. These will be in the form of Key=Value pairs. Refer to [Token Library](#) for a list of possible tokens. It can be easiest to send your customers a table with the values you need.



# Recap

- 01 Data Sharing Philosophy and Design**  
Overview of supported standards, our websites, and architecting your data exchange
- 02 Register a Client Record**  
Obtain client IDs for implementation of OAuth 2.0
- 03 Develop and Test**  
Simulate app launches and connectivity by connecting to our FHIR developer sandbox
- 04 Customer Implementation and Going Live**  
Strategize your install project and Go Live with Epic customers

# Recap

## Let's Get Started



**Step-by-Step  
Developer Guide**



**Data Sharing  
Playbooks**



**API and Interface  
Specifications**



**Need Guidance  
or Access**



## Contact Information

**Sean Hubber** | [shubber@epic.com](mailto:shubber@epic.com)

**open.epic inquiries** | [open@epic.com](mailto:open@epic.com)

# Q&A

© 2025 Epic Systems Corporation. All rights reserved. PROPRIETARY INFORMATION - This item and its contents may not be accessed, used, modified, reproduced, performed, displayed, distributed or disclosed unless and only to the extent expressly authorized by an agreement with Epic. This item is a Commercial Item, as that term is defined at 48 C.F.R. Sec. 2.101. It contains trade secrets and commercial information that are confidential, privileged, and exempt from disclosure under the Freedom of Information Act and prohibited from disclosure under the Trade Secrets Act. After Visit Summary, ASAP, Aura, Beacon, Beaker, Beans, BedTime, Best Care Choices for My Patient, Bones, Break-the-Glass, Buggy, Caboodle, Cadence, Canto, Care Everywhere, Charge Router, Cheers, Chronicles, Clarity, Cogito ergo sum, Cohort, Comfort, Community Connect, Compass Rose, Cosmos, Cosnome, Cupid, Discovery, Epic, EpicCare, EpicCare Link, Epicenter, EpicShare, EpicWeb, Epic Earth, Epic Nexus, Epic Research, Garden Plot, Grand Central, Haiku, Happy Together, Healthy Planet, Hello World, Hey Epic!, Hyperdrive, Hyperspace, Kaleidoscope, Kit, Limerick, Lucy, Lumens, MyChart, Nebula, OpTime, Phoenix, Powered by Epic, Prelude, Radar, Radiant, Resolute, Revenue Guardian, Rover, Share Everywhere, SmartForms, Sonnet, Stork, System Pulse, Tapestry, Trove, Welcome, Willow, Wisdom, With the Patient at Heart, and WorldWise are registered trademarks, trademarks, or service marks of Epic Systems Corporation in the United States of America and/or other countries. Other company, product, and service names referenced herein may be trademarks or service marks of their respective owners. Patents Notice: [www.epic.com/patents](http://www.epic.com/patents).