digital bridge

Public Health API White Paper

Version 1.0

Drafted by the Digital Bridge Public Health API Workgroup

Workgroup Chair – Walter Suarez, MD, MPH

Monday, January 18, 2021

Table of Contents

Workgroup Members

Workgroup Member	Organization	Designation
Walter Suarez	**Kaiser Permanente**	**Chair**
Indu Ramachandran	Kaiser Permanente	Member
Kirsten Hagemann	Cerner	Member
Priyanka Surio	ASTHO	Member
Ankur Jain	ASTHO	Member
Richard Hornaday	Allscripts	Member
Joe Wall	MEDITECH	Member
John Stamm	Epic	Member
Danielle Friend	Epic	Member
Ben Moscovitch	Pew Charitable Trust	Member
Ashley Ashworth	Pew Charitable Trust	Member
Molly Murray	Pew Charitable Trust	Member
Erik Knudsen	CDC	Member
Brandon Talley	CDCF	Member
Dan Chaput	ONC	Observer
Rachel Abbey	ONC	Observer

1.0 Introduction

Over the past five years there has been a major transformation in the way health care organizations exchange health information with each other and with their customers. This transformation was initiated by the widespread adoption and use of Application Programming Interfaces, commonly known as APIs, third-party apps, mobile devices, and new technical standards, such as HL7 Fast Healthcare Interoperability Resource (FHIR). APIs were originally conceived back in the 1960s when engineers needed to 'interface' end-user applications and other components to a mainframe system running applications. Modern web-based APIs started in the early 2000s when companies started to launch APIs to allow external clients to interact with their applications and data. Open APIs are now used extensively and in critical day-to-day operations in many sectors and industries of our economy, such as retail commerce, banking, airlines, and food services. Health care has been a late adopter of this open technology. However, the market has rapidly shifted, and today the majority of health care providers, health plans, and other health care organizations are embracing the use of APIs as an effective and valuable way to interact with each other, and particularly with their patients and members.

One area where APIs are beginning to have an impact and show promising opportunities is public health. Many interactions between health care providers, health plans, and community-based organizations with public health are still extensively performed today through manual processes, paper forms, and asynchronous electronic information exchanges using a variety of technical standards and proprietary or heavily customized solutions. Electronic case reporting and lab reporting, in the time of COVID-19, are perfect examples of different approaches currently in use for data reporting. As we move to more interactive, real-time, public health reporting, API technologies become more useful. To address COVID-19 related case reporting, CDC developed "eCR Now," that includes a FHIR-based reporting app that takes advantage of these new and innovative technologies to obtain faster, more reliable COVID-19 case reports that include more detailed clinical data. One key intent of the development of a common generic API Infrastructure for public health is to enable public health agencies to have direct, secure, and interactive access to up-to-date clinical information about specific patient-cases.

Considering the importance that APIs, third party apps, and HL7 FHIR will have in the future of public health reporting, the Digital Bridge Initiative chartered a working group to develop a public health API white paper. The main purpose of this white paper is to serve as a reference and provide valuable information and tools for public health professionals as they look to develop and implement their agency's or organization's public health API strategy. The white paper includes an introductory overview of APIs in general and as they apply to public health, a summary of recent health policy developments related to API, basic technical API concepts and building blocks, public health use cases, policy and privacy issues, steps needed to implement a public health API strategy, and a listing of tools and resources available to support implementation of an API strategy.

This Digital Bridge public health API white paper is intended primarily for public health professionals in local, state, and federal agencies, industry groups, and professional associations. This white paper also targets groups implementing or developing a set of common, generic API Infrastructure capabilities so

they can understand and support any capabilities or variations on capabilities needed to support public health. It is not intended to serve as a roadmap for implementation of an API platform or program. Rather, it is intended to be a resource to better understand the basic API concepts and opportunities for public health

2.0 Background

This section of the report highlights 1) policy developments in health care that foster the advancement, adoption and use of API for different health care purposes, including public health; and 2) the value proposition, benefits, and opportunities offered by APIs for public health.

2.1 Policy Developments Affecting APIs in Health Care

In the past four years there have been several important federal and state legislative and regulatory policy developments that are important for the advancement of APIs in health care and public health. These policy developments have focused on strengthening the health information infrastructure of the nation, advancing the data liquidity and access to health information, and adopting and using newer health information exchange technologies, technical standards, and functional capabilities.

Important recent federal policy developments include:

- 21st Century Cures Act of 2018, which defined the concept of Interoperability, required that health information be made available electronically to patients and consumers, called for the development of a trusted framework of health information exchanges, and prohibited information blocking.

- FY 2020 Federal Spending Legislation, which provided $50 million in new monies in support of CDC's efforts to modernize public health data systems.

- ONC Information Blocking Final Regulations, which required that certified health IT and electronic health record systems be capable of implementing and supporting API technologies and related standards. It also defined the concept of information blocking practices and provided a set of narrow, specific information blocking exceptions for health care providers, health IT vendors, health information networks/health information exchanges (HINs/HIEs).

- CMS Interoperability and Patient Access Final Regulations, which required health plans doing business with federal health programs to implement API capabilities to support member access to their health information as well as access to provider and pharmacy directories via APIs.

- 2020 CARES Act, which provided additional funding to support enhancement of public health information system capabilities to address COVID-19 reporting needs.

In addition to these federal policy developments, there have been a number of state policy developments focusing primarily on privacy of health information collected, maintained, used and disclosed by digital companies (such as the California Consumer Privacy Act), as well as enhanced cybersecurity protections and reporting requirements.

More recently, due to the global pandemic, the health care industry has rapidly moved to an ecosystem that includes more virtual care, remote monitoring and digital connectivity and exchanges, many with significant use of API technologies.

2.2 The API Value Proposition for Public Health

Overall, open APIs provide innovative, simplified, standards-based, effective, efficient, timely, and bi-directional functionality that significantly benefits public health and its interaction with external partners, and with internal stakeholders as well.

- Standardized APIs allow public health to obtain data from external stakeholders (such as providers, health plans, community-based organizations, in general – data reporters) through a more efficient, less expensive, less resource-intensive mechanism
- APIs also allow public health to seek information, in real time and from the data source, whenever that information is needed for public health to fulfill its responsibilities
- APIs also supports public health in providing more efficient, timely, and reliable mechanisms to allow access to their databases from outside stakeholders, and through those access points, allow stakeholders, when appropriate, to upload data into the registry or database.
- APIs can complement and, in several cases replace current less efficient, effective and more complex methods to collect, receive, query, access, or disclose data from and by external stakeholders
- Examples include electronic case reporting, immunization registries, health care cost/utilization databases, and others

3.0 Basic API Concepts (Building Blocks)

3.1 API Overview

The following chart provides an overview of APIs, their basic components, and their use in different data interactions.

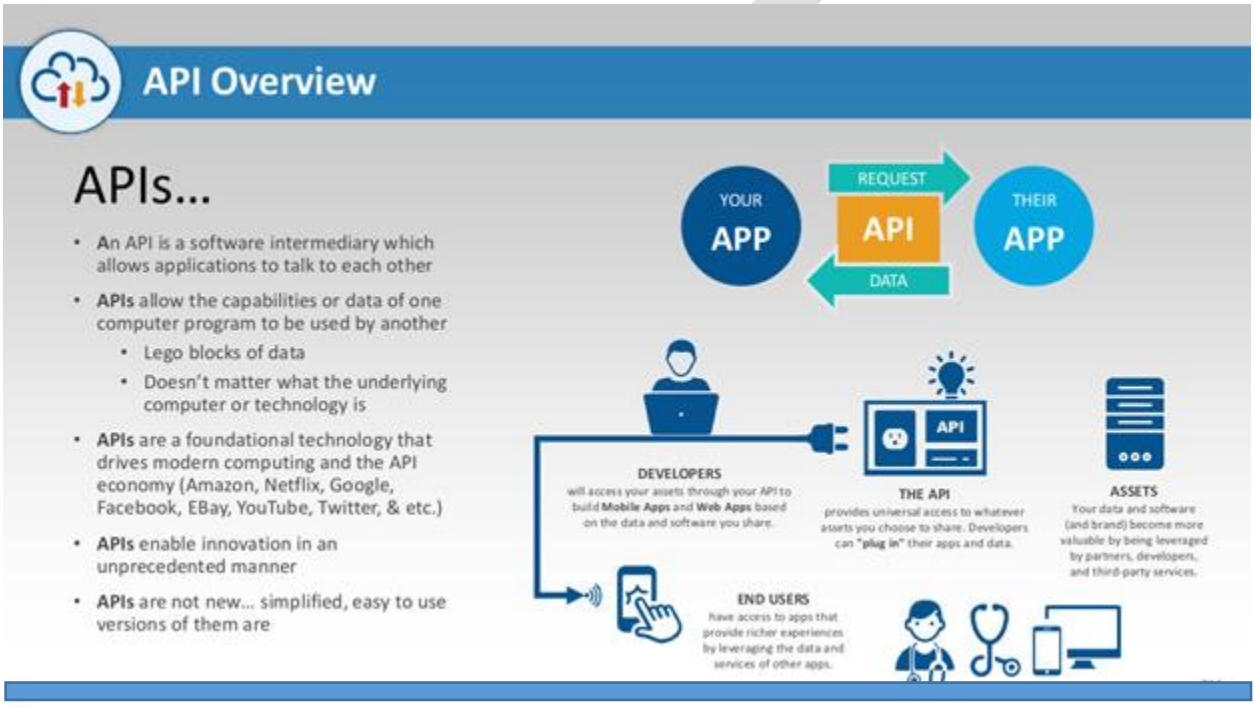

Figure 1 below provides an illustration of the basic API building blocks. The subsequent sub-sections provide more details about each of the subcomponents.

The intent of defining the API Infrastructure in this manner is to encourage the creation and maintenance of a common, generic API infrastructure that can be used for multiple Public Health use cases instead of establishing multiple 'purpose-built solutions', each with their own unique infrastructure. In addition, by leveraging common API infrastructure work being developed for activities beyond public health, the goal would be to merge and coordinate Public Health needs within general-purpose API infrastructures. In this manner, public health service delivery cost and timeliness can be optimized.

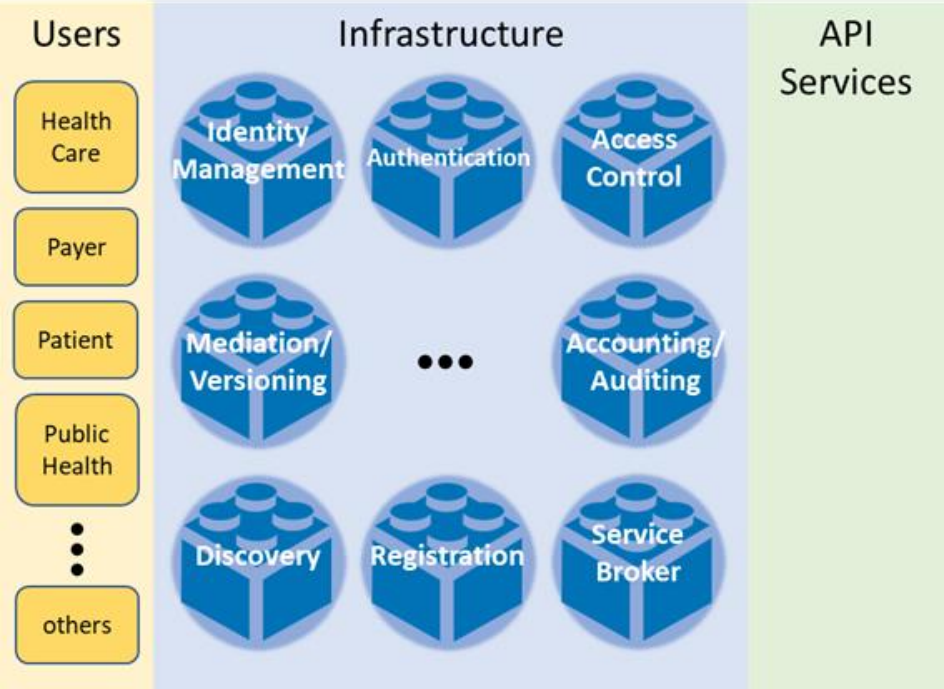

Figure 1: Public Health API Building Blocks

3.2 Users

Users are "consumers" of API resources and services. In a public health context, there are many broad classes of users, but the key user classes include health care (clinicians and other providers of health care services), health plans, patients, and public health. While these are the primary users, the intent is for the API infrastructure to be able to accommodate other user classes.

In a more generic view a User, there typically will be an *Actor* (a real person) working through an *Agent* (a system to facilitate their interactions with the API infrastructure). There are also other scenarios where the interactions are automated (i.e., not directed by a human Actor).

A generic relationship between Actors, Agents, and the API Infrastructure is shown in Figure 2 and some illustrative (non-comprehensive) examples are show in Figure 3. As show in Figure 3, the Agent systems can be very simple (e.g., a phone app) or more complex (a complex EHR system working in conjunction with a complex HIE). The key thing is to ensure that the Agent system(s) provide transparent access to the API infrastructure in a user-centered environment where the system complexity is not visible to the human Actor.

In all cases, the interactions between the Actor, the Agent(s), and the API Infrastructure is shown as bi-directional since these interactions will be very transactional.

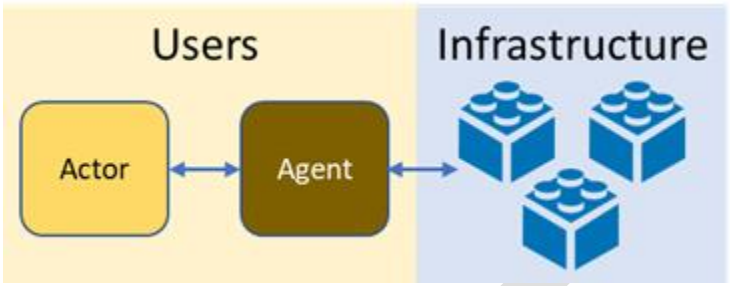

Figure 2: Users' Access to the API Infrastructure - Generic

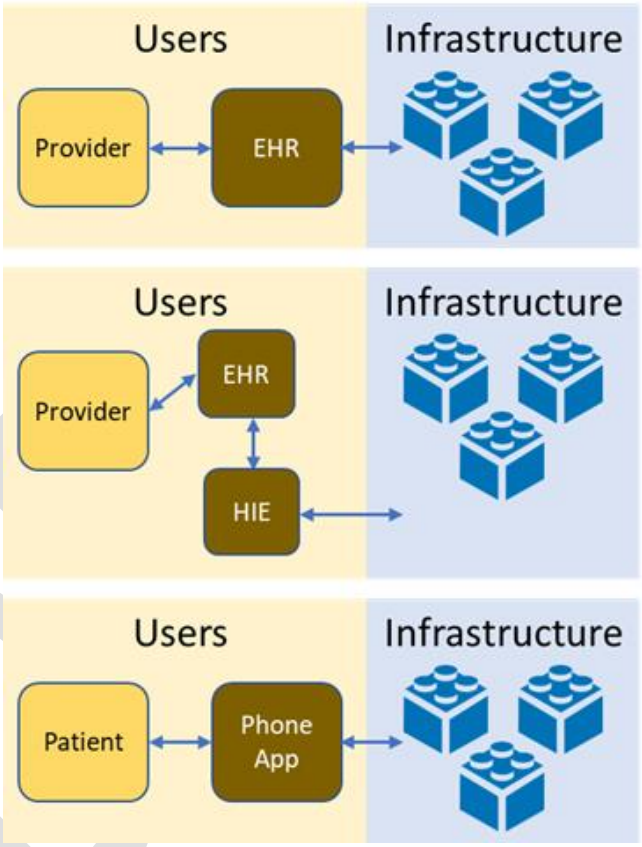

Figure 3: Users' Access to the API Infrastructure – Illustrative Examples

3.3 Infrastructure Elements

As noted previously, the ideal would be for public health leveraging common API infrastructure work being developed for activities beyond public health. As such, other work such as the ONC's FHIR at Scale Taskforce (FAST) can and should be used for common infrastructure elements in a FHIR API environment. The following highlights illustrative details where needed to deviate from "FHIR-based APIs" and any special needs for public health. It is important to emphasize that these activities must be performed the same way by all participants.

3.2.1 Identity Management

Identity Management is key for the public health context with a couple of critical factors:

- Identity Management for the Public Health actors will be critical
- Identity Management for the Providers (individuals, associations as part of a Practice)
- Identity Management for Patients, especially the need for accurate but strong patient identity matching capabilities
    - Specific to this, "Patient Matching as a Service" could be considered a key infrastructure element.

3.2.2 Security

Within an API infrastructure, there are many components that support Security, including Identity Management, Authentication, Access Control, and Accounting/Auditing. Given the sensitive nature of any personally identifiable information within public health repositories, there are clearly added requirements for public health.

- Authentication: ensuring any individual or system attempting to interact with the API infrastructure must be appropriately identified, validated and authenticated to the API application.
- Access Control: ensuring any access control infrastructure can manage data access controls delineation at granular levels to enable controls for portions/segments of data within public health repositories depending on the granted permissions of the User and/or potentially complicated consent designations that sometimes are associated with public health data.
- Accounting/Auditing for public health may need to have the capability to provide granular access control logs to ensure forensic evaluation of any access activity.

3.2.3 Registration

While this is closely coupled with security infrastructure, an API infrastructure needs to enable users to register with API services to manage their access and establish their appropriate access controls.

3.2.4 Discovery

Discovery typically refers to the ability to automatically detect devices and services on a network. For this context, Discovery also includes the ability of users to discover the rich set of API services and potentially the set of API Infrastructure elements available.

Public health historically has operated as many disparate purpose-built systems, often associated to a specific region/territory/state. As such discovery of the API Service(s) that may be available for any given user or patient for any given purpose has the potential to be quite complex.

3.2.5 Service Broker

While this is closely coupled with Discovery infrastructure, in a commercial API Infrastructure where there are potentially multiple competing API resource/service providers for a given capability/service, a Service Broker capability may be needed to ensure non-biased equal access to these capabilities/services for users that do not express any preference. A Service Broker can also be the "visible" par of a discovery service.

A variation of a service broker capability can also be used to achieve performance and scaling needs by providing effective distribution of system load across multiple supporting systems while hiding the complexity of the architecture from users.

3.2.6 De-Identification

Access controls associated with public health data repositories may require the need for de-identifying the data for some users. While some API services and repositories may already offer this as part of their handling/processing, this is a prevalent system need within public health so providing this capability as an API Infrastructure element may be useful both for improving the efficacy of the de-identification as well as the efficiency of providing this complex capability across multiple API services.

Another methodology being considered in de-identification, particularly in the context of COVID-19 data strategy, is Privacy Preserving Record Linkage (PPRL), a system of identifying and linking records that correspond to the same subject across several data sources hosted by different parties, without revealing any sensitive information about the subject.

3.2.7 Testing & Certification

While the need to ensure that any API Infrastructure element and/or an API Service is valid and meets minimal requirements is essential in any healthcare IT ecosystem, this need is heightened for Public Health due to the sensitive nature of much of its data. The Discovery and Service Broker capabilities will inherently need to be limited to only those items that have achieved the requisite levels of certification. There also will be an associated need for verification of any testing/certification claims of a given API Infrastructure element or API Service.

3.2.8 API Services

Whereas Users of the API Infrastructure and Services, the API Services provide the resources and processing to serve the multitude of health IT needs. Some API Services may themselves need to be consumers of other API Services (i.e., an API Service may assume the role of a User for other API Service(s)). This type of recursive use of the API infrastructure is a natural and important aspect of ensuring efficiency.

4.0 Use Cases

4.1 API Use Cases in Healthcare

There has been a need for modern technology in order to provide a more connected healthcare experience for all stakeholders such as patients, providers, health plans, medical devices and public health.

APIs will allow for technology in healthcare to reach to the level of other industries to access information from disparate systems and networks such as the banking or travel industry where an application can gain access to information that was previously siloed.

4.2 Use case definition

An API can enable a system to send or retrieve data that can update an individual's record or provide collected data that can be used to create reports. It is important to distinguish these two applications of API – sending and retrieving data. Generally, organizations can use APIs to send (or receive) data – for example, updating records in the target system. While this is used in some instances, it does involve certain levels of risk and is generally not widely implemented today. More common is the use of APIs to allow external sources to retrieve data from the host, in a secure manner.

There is typically a requestor of information and then there is a receiver of information. The user is typically using an app that can receive or provide information to others that can be accessed through another system and can interpret that information as if they were on their own native system.

APIs are typically reliant on an endpoint where requesters can query specific information in a known format.

Prior to this ability, users had to rely on phone, fax to get information not readily in the system. It's a common language where all systems can speak. There have been strides to advance technology to reduce the amount of manual work in order to be automated. EHR Incentives such as Meaningful Use/Promoting Interoperability have introduced and adopted standards in the public health space, such as HL7 v2 (Immunization Registries, Syndromic Surveillance and Reportable Labs), CDA (Cancer Registries and eCR), and HL7 FHIR.

4.3 Health Care Examples

4.3.1 Patient-facing APIs

Patient health Information can be fragmented, and patients may have records stored on their primary care physician's system as well as information stored within their local hospital which could be on a different EHR system.

By virtue of an app utilizing API technology, this use case can be used to aggregate personal health records across disparate systems. This is important when looking at population-level analysis of data and can help address the challenge of having multiple portals from different sources (providers, payers, others) where data about the same individual resides.

In the future, and to some extent today, patients can also leverage APIs to capture Patient Generated Health Data (PGHD) such as blood pressure and other vital signs, glucometer devices, so that such data can be incorporated into a provider's system and be acted upon.

APIs are also increasing efficiencies in scheduling appointments, sharing questionnaires electronically and paying bills right from a patient's own device, via an API interface hosted by the health care organization.

There is also a regulatory requirement for Promoting Interoperability (formerly known as Meaningful Use) for Patient Electronic Access. 21st Century Cures/CMS Interoperability and Patient access will require all healthcare organizations to adopt this technology so everyone is on the same level playing field.

Other areas where a patient could utilize an app to streamline processes to direct their health information to whoever needs it.

4.3.2 Provider-facing APIs (CDS, others)

APIs will facilitate the ability to view and access information in another vendor's system. APIs can be used along other protocols to search and retrieve records from disparate sources. CommonWell Health Alliance is already utilizing APIs in this fashion. There are a number of common use cases in public health for provider-facing APIs, such as patient referrals, laboratory testing, transitions of care, public health case reporting, immunization registries, and others.

APIs such as FHIR have the ability to query and exchange smaller pieces of information as requested. Traditionally, interoperability between systems has been based on a document-exchange approach (for example, using HL7 V2 messages – and the volumes remain high today). One of the barriers towards interoperability is physicians reviewing multiple documents to obtain the information that they are interested in. A FHIR based API has the potential to greatly reduce the amount of document exchange occurring and only exchange the items that are of clinical significance to the physician/clinician.

APIs can also be used to have EHR systems interact with 3rd party rules such as CDS (Clinical Decision Support) based on triggers from information entered within the EHR. SMART on FHIR is a current technology that can be leveraged for this.

4.3.3 Others (Health Plan applications, Bulk Data, administrative functions)

Recent regulations will require the support of APIs from a health plan perspective so that both patients and providers can access claims information to eliminate blind spots so that they are getting the entire patient story.

There is also a need to offer the same capability to deliver population-level data to providers. FHIR Bulk Data will allow the exchange of large volumes of patient data from external platforms and present in ways that are intuitive and meaningful.

MyHealthEData and BlueButton+ are current initiatives that are moving this along.
FHIR Bulk data has the potential to transform and drive population health

APIs have the potential to improve clinical research by allowing more patients/subjects to participate. APIs can retrieve records on many patients based on certain criteria while allowing the records to remain anonymous.

4.4 Public Health Examples

Public health is already utilizing APIs in a number of areas, including:

- EHR retrieving historical and forecast information from an **immunization registry**
- **Syndromic Surveillance** - Provides public health officials with timely information in order to detect and monitor health trends.
- **IHE PRQ** (IHE Prescription Repository Query)

- COVID 19-related use cases
  - **eCR Now** - an open source FHIR based application for rapid adoption of electronic case reporting specifically for COVID-19 reporting.
  - **ELR** (Electronic Reportable Labs) - Used to improve the reporting of notable conditions to Public Health. It is currently a Promoting Interoperability/Meaningful Use objetive based on HL7 2.5 standards.

- **SANER** (Situational Awareness for Novel Epidemic Response)- developing a standardized capacity reporting system

The use of APIs can spur innovation and new technology applications. Systems can continue to be developed, expanded and updated in order to become an open environment in which applications can be integrated into any systems and be utilized as a platform. This could enhance an organization's business strategy.

An example of this is SMART on FHIR which is built on **SMART Health IT** to allow EHR systems to become a platform to be utilized by 3rd party applications.

4.4.1 Future Work

The outbreak of COVID-19 has identified areas where API technology could fit in, particularly with the need for timely reporting to agencies (CDC, public health) for capacity reporting as well as COVID-19 positivity reporting.

This reporting to date has relied on a manual approach where reports are generated out of the electronic health record and then uploaded to a file server. There is an opportunity to use APIs to better automate and reduce the burden on the healthcare organizations and receivers of information, so that relevant data (individual or aggregated) can be queried.

Reporting requirements to date, as outlined in the U.S. Department of Health and Human Services (HHS) require hospitals that conduct COVID-19 detection testing and/or antibody testing, to report daily aggregate result data to HHS. In addition, Condition of Participation (CoP) requires Medicare and Medicaid participating hospitals and critical access hospitals to report COVID-19 testing, capacity, and utilization data to the Department of Health and Human Services (HHS). APIs could introduce efficiencies whereby HHS could query, in a standardized format, from the hospital system.

APIs and related standards, such as HL7 FHIR are promising in this space as it has been embraced by most electronic health record systems.

Current work, such as eCR Now has leveraged this technology and can be used for other use cases (as mentioned above) for more scale.

5.0 Policy, Privacy, and Public Health

This section of the White Paper focuses on different policy and privacy considerations when implementing API functionality for purposes of public health information exchanges. To identify and address these policy and privacy considerations, it is important to note a few critical elements related to information exchange.

There are federal laws and regulations that define which data is required to be reported when, how and to whom in the federal government (for example CDC), for certain public health purposes (such as COVID-19). In addition, there are federal laws and regulations that affect the way in which health information is exchanges with public health agencies, most notably, the Health Insurance Portability and Accountability Act (HIPAA) and related privacy regulations.

  - Generally, these federal laws and regulations allow entities subject to them to disclose individually identifiable health information, without authorization, to a public health authority (such as a local or state public health agency or their designated agents) who is legally authorized to receive such information for the purpose of preventing or controlling disease, injury, or disability. This would include the reporting of a disease, injury, or condition; reporting vital events, such as births or deaths; and conducting public health surveillance, investigations, or interventions.
  - For disclosures to a public health authority, covered entities may reasonably rely on a minimum necessary determination made by the public health authority in requesting the protected health information.
  - There are other public health related functions for which these laws and regulations permit the disclosure of individually identifiable health information to a public health authority, including child abuse and neglect, quality, safety or effectiveness of a product or activity regulated by the FDA (such as collecting or reporting adverse events, tracking FDA-regulated products, enabling product recalls and conducting post-marketing surveillance), persons at risk of contracting or spreading a disease, and workplace medical surveillance.
  - There are federal laws and regulations that define when and how federal public health agencies may disclose or make available certain data to others.

- There are state laws and regulations and local ordinances that define what data is to be reported when, how and to whom in the state (for example state or local public health agencies), for certain public health purposes. The complexity in state, local, territorial, and tribal legal differences, and the complexity introduced to any national effort or approach, governmental or technical cannot be understated.

- There are also state laws and regulations and local ordinances that define when and how public health agencies in the state may disclose or make available certain data to others.

A number of these federal and state laws and regulations do specify the method or mechanism for external stakeholders to report data to the public health authority, such as paper forms, faxes, or electronic documents using identified standards. There are now some considerations for adopting and using APIs as a method for allowing the reporting of public health data, or for public health authorities to access from the source data that they need in order to perform their functions.

When it comes to public health authorities making data available to external stakeholders, different methods exist as well, including electronic documents structured in different formats. APIs are now being used by public health authorities as another method to make data available externally. See examples of these use cases in Section 4.0.

In this report we look at three scenarios to help identify and address policy and privacy considerations of using APIs.

5.1 Submission of Individually Identifiable Health Information to Public Health via API

In this scenario, the public health authority uses an API as a platform for external sources to submit the data that is required in order to fulfill the program expectations. One example is an Immunization Registry where a provider submits data via an API. Is important to note that in some instances under this scenario (for example, bulk data submissions of multiple data points about multiple individuals), an API might not be the most efficient method for collecting/reporting data.

- Would there be a need for a federal or state action to adopt the use of API as a method to be used by external data sources to fulfill public health reporting requirements? For example, if a state wants to implement an API in their prescription drug monitoring program, electronic case reporting, syndromic surveillance systems, specialty registries (such as immunization registry), or other programs, would they need to define this via a policy action? It is not clear whether there is a need for such legislative or regulatory actions. Rather, defining a specific submission method might be an action at the program level that may be best handled through program requirements.

- Would there be any distinct privacy considerations related to the use of an API platform to allow external sources to submit data to public health authorities – beyond what already exists in federal and state privacy laws and regulations? Submitters of data would need to follow the same privacy policies and requirements that already exist. New security considerations (such as authentication, authorization, access controls and audit trails at the submitter and receiver points) would need to be established for the new API platform.

5.2 Public Health Authority Seeking to Access Data from External Sources via API

In this scenario, the public health authority seeks specific data about specific individuals from an external source (such as a provider) using an API platform available from the data source, with all appropriate security controls. For example, public health needs to obtain additional clinical information about a case under investigation, and instead of requesting the data and waiting for a response, the public health authority accesses the data from the source.

- Would there be a need for a federal or state action to allow a public health authority to access data via an API from an external source? In some instances, in today's practice, when a public health authority receives a case report from a provider, the public health authority requests a copy of additional clinical information, and sometimes sends a person to obtain the information from a provider where the data might be. Obtaining such information electronically, remotely, via an API would not be that much different from doing it manually – except for the security controls needed to ensure appropriate access and audit controls. Under HIPAA the provider may reasonably rely on a minimum necessary determination made by the public health authority in requesting access to the data.

- Would there be any distinct privacy considerations related to a public health authority accessing health information from an external source via an API provided by the source? If public health authority is accessing data via API, the source (e.g., provider) would need to establish a mechanism to securely allow such access, and to present specific types of data to the public health authority based on the query being presented through the API. Being able to parse/select the type of data that is needed to be presented to public health will be important.

5.3 External access to public health data via API

In this scenario, the public health authority uses an API as a platform to allow external stakeholders to access selected data they are making available, based on defined conditions and criteria. There would be two variants of this scenario, depending on whether the data being made available is individually identifiable health information or not.

5.3.1 External access to individually identifiable data from public health via API

An example of this scenario is a provider accessing data about a specific individual from an immunization registry via an API.

- Would there be a need for a federal or state action to allow a public health authority to make individually identifiable health information via an API from an external source? Outside of the need to have policy action to make individually identifiable health information available per-se, the method of using an API platform to achieve such access would not seem to create a need for specific policy actions. There is going to be a need to establish strong technical, physical and administrative safeguards to ensure that only authorized individuals are accessing only the right information about the right person (measures such as an API access and endpoint, authentication, authorization, access control and audit trails). New terms and conditions related to the agreements that govern access to such registries will also be needed.

- Would there be any distinct privacy considerations related to a public health authority using an API platform to allow access to individually identifiable health information? For existing use cases – such as an immunization registry – the new method of using API to allow access is not likely to carry any new or different privacy implications distinct from the ones already in place. For new use cases, there would be a need to develop appropriate privacy protections.

5.3.2 External access to aggregate (non-individually identifiable) data by public health via API

An example of this scenario is an external entity accessing a public health database with no individually identifiable health information, such as population health data from a specific area in a state, via API for analytic purposes. It is unlikely that the adoption and use of API technologies to make aggregated public health data available would engender the need for new federal or state policy or privacy actions, outside of the ones already used when such data disclosures are done today (such as prohibiting any attempts to re-identify data). Yet, this is one of the areas where APIs could be most valuable and useful to public health and to external data users, and it is a method that is being used in a number of federal programs (such as healthdata.gov) as well as state data programs.

6.0 Strategies and Steps Needed to Implement an API Approach

This section of the paper focuses on a series of suggested planning and design elements, characteristics and steps that public health organizations should consider when developing and implementing an API enterprise strategy.

A first consideration is for the organization to assess the level of readiness and pre-implementation steps that need to be taken when planning for an API strategy, including

- Leadership buy-in
- Assessment of internal capacity and resources
- Policy assessment (any barriers with existing internal or external policies to consider in working with APIs)
- State of data modernization – How planning for and budgeting for data modernization could include exploring the utility and adoption of public health API use

6.1 API design

Most modern web applications expose APIs that clients can use to interact with the application. A well-designed web API should aim to support:

- **Platform independence**. Any client should be able to call the API, regardless of how the API is implemented internally. This requires using standard protocols and having a mechanism whereby the client and the web service can agree on the format of the data to exchange.
- **Service evolution**. The web API should be able to evolve and add functionality independently from client applications. As the API evolves, existing client applications should continue to function without modification. All functionality should be discoverable so that client applications can fully use it.

6.2 Organize the API around resources

When considering a public health API strategy, the organization should focus on the business entities that the API(s) will be used for, the data the public health agency will make available via an API, and the public health systems that will be made available for interaction with both internal and external

stakeholders via an API. Examples include case reporting, lab reporting, immunizations, specialty registries, and others. Creating an eCR, for example, can be achieved by sending a "POST" request (see below) that contains the case information via a secure internet connection using web-based HTTP protocol. The web response indicates whether the case was reported successfully or not. When possible, resource URLs should be based on nouns (the resource) and not verbs (the operations on the resource).

6.3 Define operations in terms of Hypertext Transfer Protocol (HTTP) methods

The internet-based HTTP protocol defines a number of methods that assign semantic meaning to a request. The common HTTP methods used by most RESTful web APIs are listed below. RESTful refers to the Representational State Transfer, the de-facto standard of web software architecture for API-based interactive applications that typically use multiple web services. The importance and value of this characterization is that it provides a clear understanding of the types of interactions supported by APIs and helps guide the possible uses that public health can give to APIs for its different data systems.

- **GET** retrieves a representation of the resource at the specified URI. The body of the response message contains the details of the requested resource.
- **POST** creates a new resource at the specified URI. The body of the request message provides the details of the new resource. Note that POST can also be used to trigger operations that don't actually create resources.
- **PUT** either creates or replaces the resource at the specified URI. The body of the request message specifies the resource to be created or updated.
- **PATCH** performs a partial update of a resource. The request body specifies the set of changes to apply to the resource.
- **DELETE** removes the resource at the specified URI.

6.4 Maintaining responsiveness, scalability, and availability

The same web API might be used by many client applications running anywhere in the world. It is important to ensure that the web API is implemented to maintain responsiveness under a heavy load, to be scalable to support a highly varying workload, and to guarantee availability for clients that perform business-critical operations. Consider the following points when determining how to meet these requirements.

6.5 Provide asynchronous support for long-running requests

A request that might take a long time to process should be performed without blocking the client that submitted the request. The web API can perform some initial checking to validate the request, initiate a

separate task to perform the work, and then return a response message that its was accepted. The task could run asynchronously as part of the web API processing, or it could be offloaded to a background task.

The web API should also provide a mechanism to return the results of the processing to the client application. Organizations can achieve this by providing a polling mechanism for client applications to periodically query whether the processing has finished and obtain the result or enabling the web API to send a notification when the operation has completed. Organizations can implement a simple polling mechanism by providing a *polling* URI that acts as a virtual resource.

6.6 Secure API management

Build trust and engagement with secure assets in the organization's API program — at all points of engagement from users, apps, developers, API teams, and backend systems. Provide protection against hackers, bots, and other suspicious behaviors. For example, verify API keys at runtime, generate OAuth tokens, implement JSON threat protection, and more with policies that extend the built-in security layer of the API management tool.

6.7 Testing a web API

A web API should be tested as thoroughly as any other piece of software. The organization should consider creating unit tests to validate the functionality.

The nature of a web API brings its own additional requirements to verify that it operates correctly. The organization should pay particular attention to the following aspects:

- Test all routes to verify that they invoke the correct operations. Be especially aware of HTTP status code 405 (Method Not Allowed) being returned unexpectedly as this can indicate a mismatch between a route and the HTTP methods (GET, POST, PUT, DELETE) that can be dispatched to that route.
- Send HTTP requests to routes that do not support them, such as submitting a POST request to a specific resource (POST requests should only be sent to resource collections). In these cases, the only valid response *should* be status code 405 (Not Allowed).
- Verify that all routes are protected properly and are subject to the appropriate authentication and authorization checks.

Some aspects of security such as user authentication are most likely to be the responsibility of the host environment rather than the web API, but it is still necessary to include security tests as part of the deployment process.

- Test the exception handling performed by each operation and verify that an appropriate and meaningful HTTP response is passed back to the client application.
- Verify that request and response messages are well-formed. For example, if an HTTP POST request contains the data for a new resource in x-www-form-urlencoded format, confirm that the corresponding operation correctly parses the data, creates the resources, and returns a response containing the details of the new resource, including the correct Location header.
- Verify all links and URIs in response messages. For example, an HTTP POST message should return the URI of the newly created resource
- Ensure that each operation returns the correct status codes for different combinations of input. For example:
    - If a query is successful, it should return status code 200 (OK)
    - If a resource is not found, the operation should return HTTP status code 404 (Not Found).
    - If the client sends a request that successfully deletes a resource, the status code should be 204 (No Content).
    - If the client sends a request that creates a new resource, the status code should be 201 (Created).

Watch out for unexpected response status codes in the 5xx range. These messages are usually reported by the host server to indicate that it was unable to fulfill a valid request.

- Test the different request header combinations that a client application can specify and ensure that the web API returns the expected information in response messages.
- Test query strings. If an operation can take optional parameters (such as pagination requests), test the different combinations and order of parameters.
- Verify that asynchronous operations complete successfully. If the web API supports streaming for requests that return large binary objects (such as video or audio), ensure that client requests are not blocked while the data is streamed. If the web API implements polling for long-running data modification operations, verify that the operations report their status correctly as they proceed.

The organization should also create and run performance tests to check that the web API operates satisfactorily under duress.

6.8 Publishing and managing a web API

To make a web API available for client applications, the web API must be deployed to a host environment. This environment is typically a web server, although it may be some other type of host process. The organization should consider the following points when publishing a web API:

- All requests must be authenticated and authorized, and the appropriate level of access control must be enforced.
- A commercial web API might be subject to various quality guarantees concerning response times. It is important to ensure that host environment is scalable if the load can vary significantly over time.
- It may be necessary to meter requests for monetization purposes.
- It might be necessary to regulate the flow of traffic to the web API, and implement throttling for specific clients that have exhausted their quotas.
- Regulatory requirements might mandate logging and auditing of all requests and responses.
- To ensure availability, it may be necessary to monitor the health of the server hosting the web API and restart it if necessary.

It is useful to be able to decouple these issues from the technical issues concerning the implementation of the web API. For this reason, consider creating a **façade**, running as a separate process and that routes requests to the web API. The façade can provide the management operations and forward validated requests to the web API. Using a façade can also bring many functional advantages, including:

- Acting as an integration point for multiple web APIs.
- Transforming messages and translating communications protocols for clients built by using varying technologies.
- Caching requests and responses to reduce load on the server hosting the web API.

6.9 Developer Portal

Customize a developer portal for a uniquely branded experience — whether deployed on the API management platform in the cloud or on-premises. Self-service access to a secure developer portal is key to successful communication.

6.10 Monitoring a web API

Depending on how the organization have published and deployed its web API, it can monitor the web API directly, or it can gather usage and health information by analyzing the traffic that passes through the API Management service.

6.11 Monitoring a web API directly

If the organization has implemented its web API by using the ASP.NET Web API template (either as a Web API project or as a Web role in an Azure cloud service) and Visual Studio 2013, the organization can gather availability, performance, and usage data by using ASP.NET Application Insights. Application Insights is a package that transparently tracks and records information about requests and responses when the web API is deployed to the cloud; once the package is installed and configured, the

organization does not need to amend any code in its web API to use it. When it deploys the web API to an Azure web site, all traffic is examined, and the following statistics are gathered:

- Server response time.
- Number of server requests and the details of each request.
- The top slowest requests in terms of average response time.
- The details of any failed requests.
- The number of sessions initiated by different browsers and user agents.
- The most frequently viewed pages (primarily useful for web applications rather than web APIs).
- The different user roles accessing the web API.

6.12 Using OAuth 2.0 to access API

All applications follow a basic pattern when accessing an API using OAuth 2.0. At a high level, the organization should consider following there five steps:

1. Obtain OAuth 2.0 credentials from the API Console.
2. Obtain an access token from the Authorization Server.
3. Examine scopes of access granted by the user.
4. Send the access token to an API.
5. Refresh the access token

6.13 App's connection to the API

1. Download the app
2. Resources Owner authenticates to authorization server
3. Authorization code is granted
4. Authorization server redirects resources owner back to the Client with an authorization code
5. Client send authorization code to the authorization server's token endpoint
6. Client authenticates using credential
7. Authorization server issue's an OAuth access token to the client
8. Access the data using the access token

7.0 Tools, Resources and References Available to Support Implementation of an API Strategy

This section of the report highlights a series of API-related tools, resources and references that will be useful to public health agencies in their consideration of the development and implementation of an API solution for their multiple information exchange needs.

7.1 Tools and Resources

- U.S. GSA API Developer Program – A service of GSA, it provides a wealth of tools and resources to help an organization's API efforts - http://18f.github.io/API-All-the-X/
- API.Data.Gov – API tools and documentation from various federal agencies - https://api.data.gov/
- Resources.Data.Gov/API – A wealth of API Resources with case studies, examples and best practices - https://resources.data.gov/keywords/api/
- HealthData.Gov – The HealthData.gov API is used to provide software developers programmatic access to the contents of the data catalog. **https://healthdata.gov/api**
- HHS Developers' Center – provides access to API-based data resources and code to access HHS Open Data - **https://www.hhs.gov/web/developer/index.html**
- ONC API Resource Guide: https://www.healthit.gov/sites/default/files/page/2020-11/API-Resource-Guide_v1_0.pdf
- ONC API Learning Modules - https://www.healthit.gov/topic/patient-access-to-medical-records/learning-module-apis-and-health-data-sharing
- CDC API Portal - **https://www.cdc.gov/apis.html**
- CDC Open Technology Developer Community - https://open.cdc.gov/code.html
- CDC eCR Now: **https://ecr.aimsplatform.org/**
-

- CDC Surveillance Data Strategies: https://www.cdc.gov/surveillance/surveillance-data-strategies/data-IT-transformation.html

- CSTE:
  - https://www.cste.org/blogpost/1084057/338009/Congress-Funds-50-Million-to-Modernize-Public-Health-Data-Systems-and-Boosts-CDC-s-Budget
  - https://resources.cste.org/data-superhighway/mobile/index.html

- CMS API Portal - https://developer.cms.gov/#apis

- NIH AP Portal – https://api.data.gov/docs/nih/

- Science.Gov – Inventory of API Resources in the Federal Government - https://www.science.gov/servicesandtools.html

7.2 References

- ONC Application Programming Interfaces in Health IT - https://www.healthit.gov/buzz-blog/21st-century-cures-act/application-programming-interfaces-in-health-it

- ONC – Understanding Emerging API-based Standards - https://www.healthit.gov/isa/understanding-emerging-api-based-standards

- ONC Key Privacy and Security Considerations for Healthcare APIs - https://www.healthit.gov/sites/default/files/privacy-security-api.pdf

- NITS Guide to Secure Web Services - https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf

- CSTE Driving Public Health in the Fast Lane - https://cdn.ymaws.com/www.cste.org/resource/resmgr/pdfs/pdfs2/Driving_PH_Print.pdf

- HLN - https://esmed.org/MRA/mra/article/view/2282/193545704

8.0 Conclusions and Next Steps

As we move to more interactive, real-time, digital public health reporting, API technologies have become more prevalent and critical. This Public Health API White Paper Version 1.0 provides an initial set of background information, basic API concepts, use case examples, policy considerations and tools and resources for public health professionals interested in learning more about developing and implementing API solutions in their environment.

The intent of Digital Bridge with respect to this paper is to make it a living document that can be expanded and updated periodically, as more examples of API uses in public health come alive.

We make three recommend next steps:

- Given the significance of the impact that APIs are having and will continue to have in Public Health, we call for the establishment of a Public Health API Community, to serve as forum for discussion, exchange of ideas and experiences, planning, identification of best practices, requirements analysis, and input about where API can be applied within public health

- We believe it is critical to develop a more formal, detailed, common, generic API Infrastructure framework for Public Health, and recommend that this be considered for action by Digital Bridge, in conjunction with the proposed Public Health API Community noted above. In this regard, we believe that this White Paper is a first phase in the development of more comprehensive documentation, tools and resources that will ultimately result in the development of such a Public Health API Infrastructure.

- We acknowledge the need to obtain more extensive input from a larger and more representative group of individuals and organizations in public health. Thus, we offer that this Version 1.0 of the White Paper be widely circulated for review and comments by the public health and health care industry at large.