Child Health PubSub

Architecture Discussion

Mike Kelly

Draft - Published 10 November 2016

**Contents**

[1.0 Introduction 3](#_Toc468954235)

[1.1 Background 3](#_Toc468954236)

[2.0 Events Management Service and PubSub 5](#_Toc468954237)

[3.0 PubSub Architecture 7](#_Toc468954238)

[3.1 Subscription Model 7](#_Toc468954239)

[3.2 Routing 8](#_Toc468954240)

[3.3 Topology 9](#_Toc468954241)

[4.0 Client Interaction 11](#_Toc468954242)

[4.1 Publisher 11](#_Toc468954243)

[4.2 Subscriber 11](#_Toc468954244)

[5.0 Child Health PubSub Considerations 12](#_Toc468954245)

[5.1 Subscription Model 13](#_Toc468954246)

[5.2 Routing 14](#_Toc468954247)

[5.3 Topology 14](#_Toc468954248)

[5.4 Client Interaction 15](#_Toc468954249)

# 1.0 Introduction

This document presents a discussion on the function, form and viability of different architectures for implementing pubsub for Child Health and the wider health and social care domains.

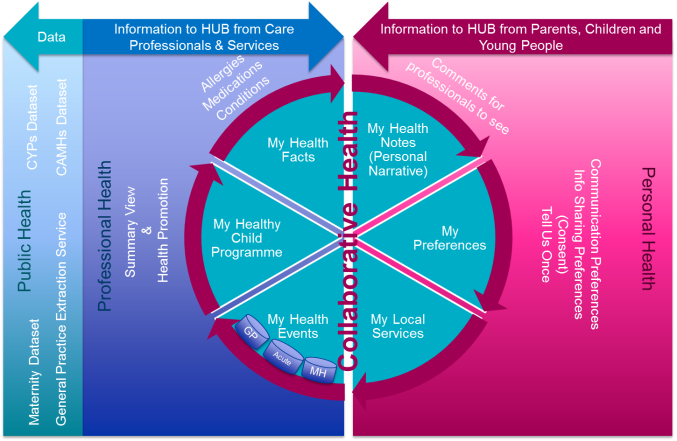
## 1.1 Background

The Children’s Health Digital Strategy is commissioned by NHS England and was published Q4 2016.

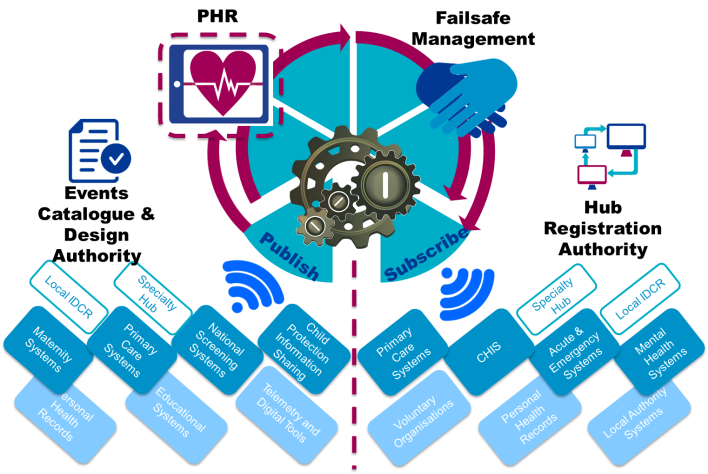
It sets out the case for transforming the way information is handled in children’s health services. It suggests that making health information interoperable – translating it into messages which can be exchanged – and ensuring that we effectively manage the offer and uptake of preventative programmes of care is the best way of achieving the key objectives of:

* Knowing where every child is and how healthy they are
* Giving appropriate access to information for all involved in the care of children

The strategy contains a vision for a Digital Child Health Hub:



The strategy also defines a set of Digital Services that are intended to be delivered, which are based on an Events Hub model:



The Digital Services are:

1. **Child Health Events Catalogue** - Child Health do not need to exchange whole care records only small event messages so the information can flow to whoever needs it in the eco system in real time. There will be multiple event types; transactional events, clinical events, consent events.
2. **Events Management Service** - A service that facilitates the exchange of messages between publishers of event information and subscribers to that information in real time.
3. **Failsafe management of the child population** - Event information and rules service, will manage both population tracking and population screening. Will tell systems location and which professionals/agencies a child is under the care of. Will prompt parents and professionals to attend for/deliver the standard programme of care and will alert if not delivered/undertaken. Needed to prevent children missing care.
4. **Digital Personal Child Health Record (ePCHR)** - Child-centric untethered PHR, needs to meet the current standards laid down by the Royal College of Paediatrics and Child Health for what constitutes an ePCHR. Significant strategic opportunity to advance self-care agenda and to join up maternity and new-born healthcare as per Maternity review by extending timeline for PHR backwards to include mother’s health.

The responsibility to deliver these Digital Services has been given to NHS Digital via the Digital Child Health Programme, which is a NIB Domain G (Paper Free at Point of Care) programme.

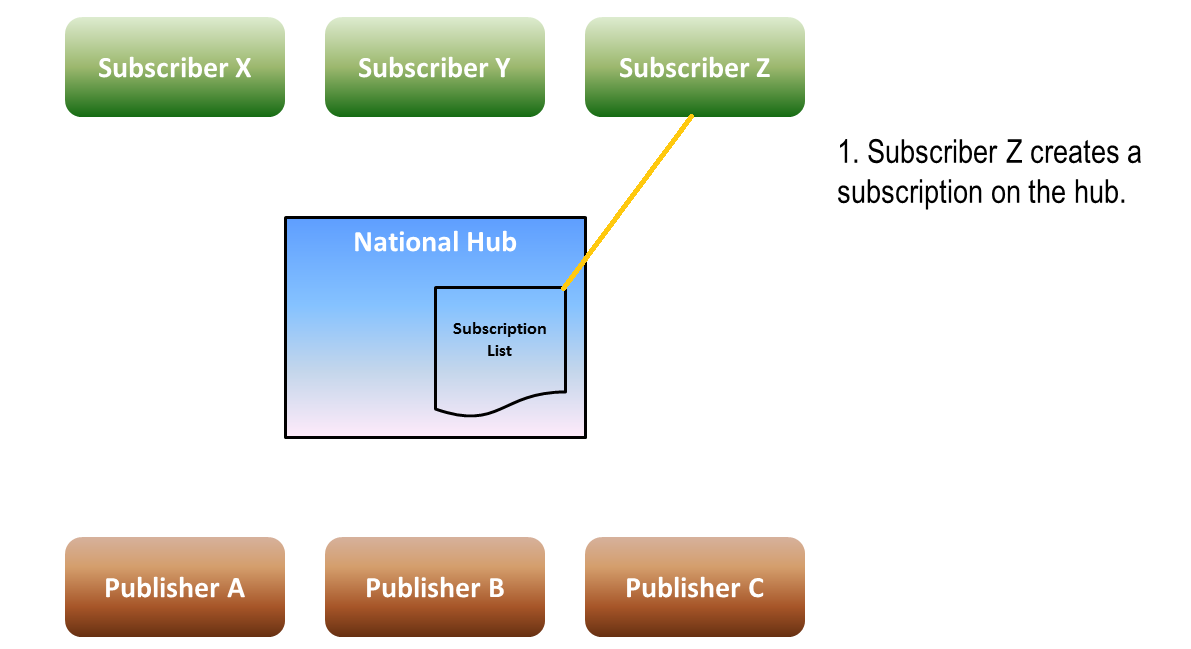
# 2.0 Events Management Service and PubSub

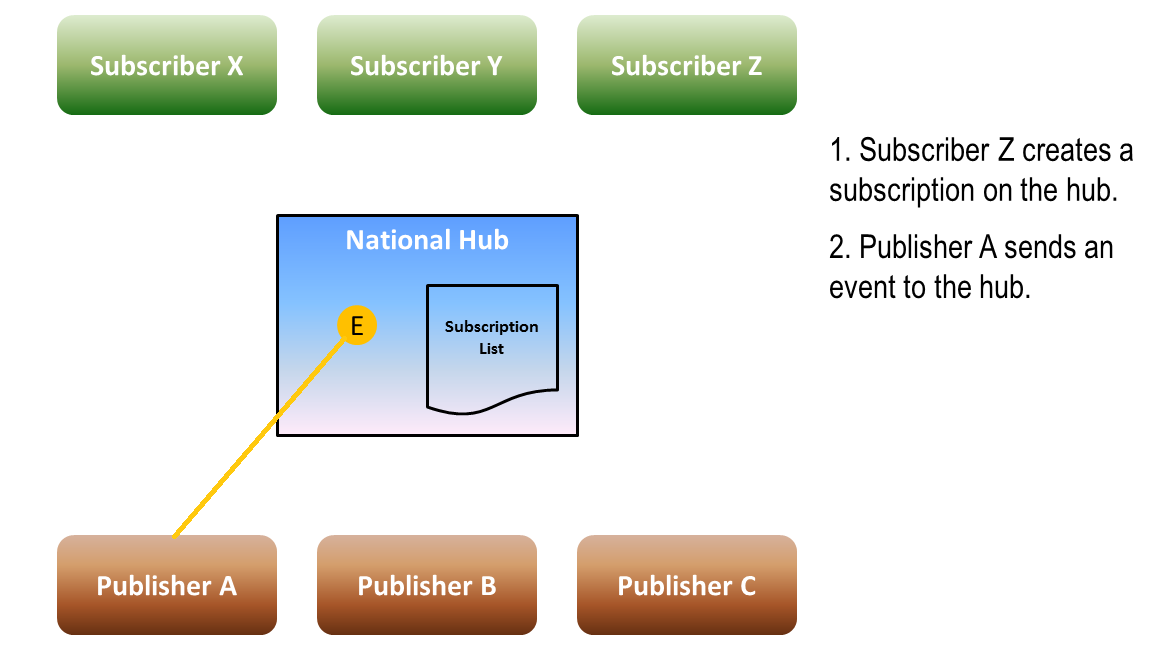
The Events Management Service facilitates the exchange of messages between publishers of event information and subscribers to that information in real time. It uses the information sharing pattern of Publication and Subscription, or PubSub, and is based on an Event Driven Interoperability Model.

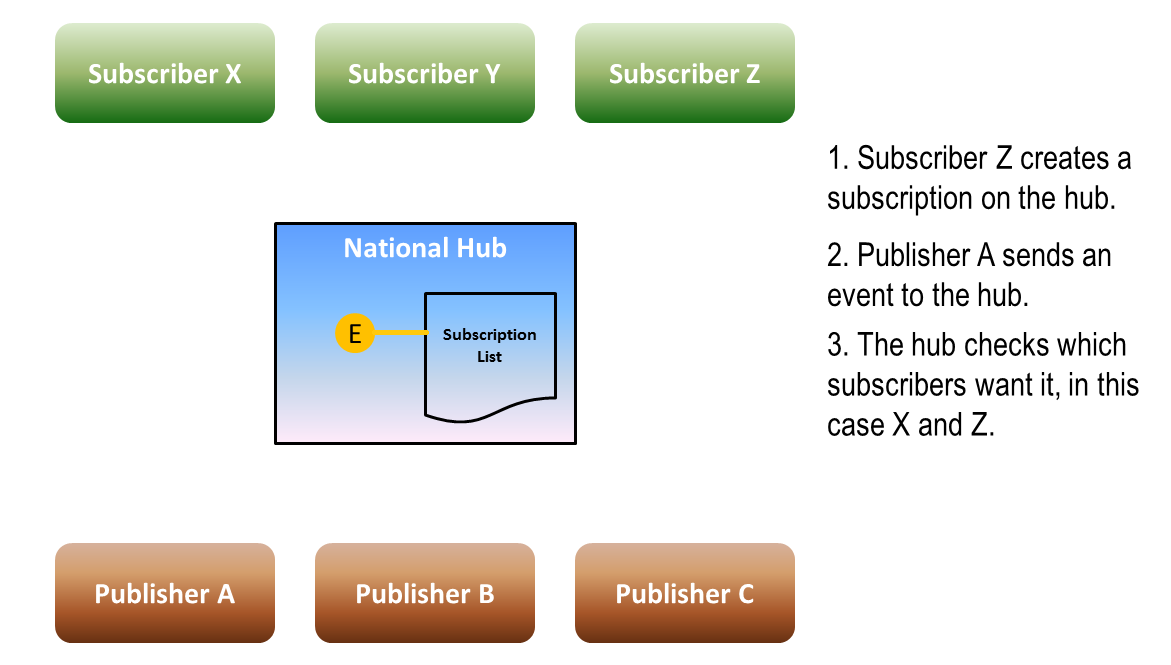
Events (small parcels of information) are sent (published) by the ICT systems used by service providers in real time to a national hub run by the NHS.

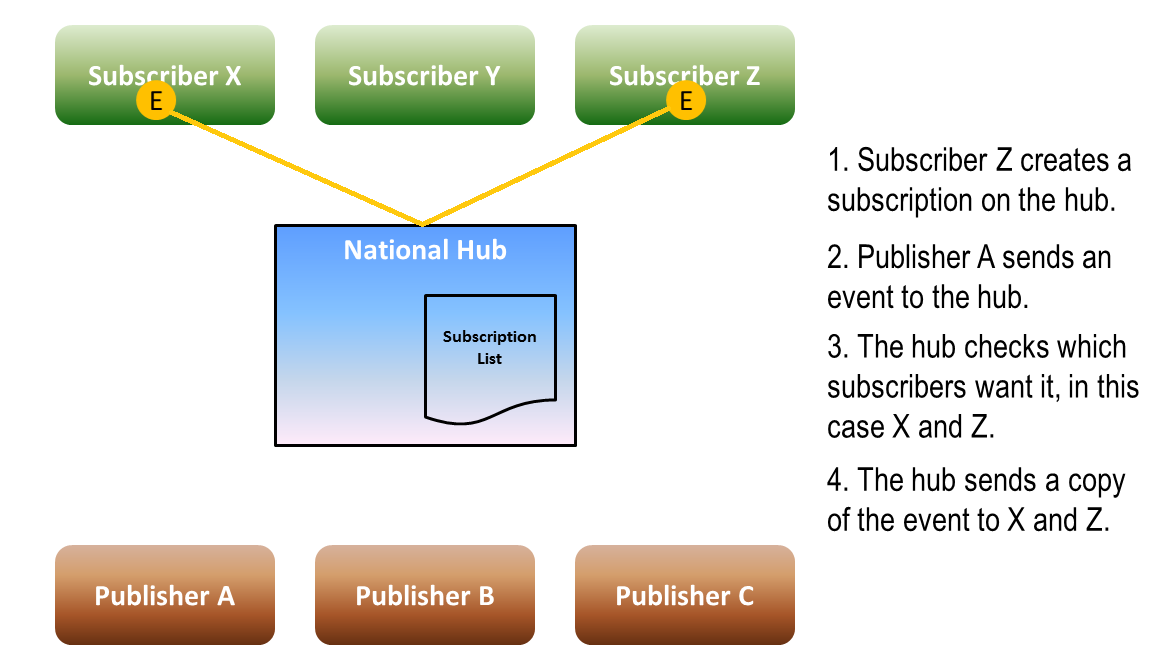
Other ICT systems that have a need for these events create subscriptions on the hub defining what types of event they want.

When the hub receives an event from a publisher it matches it to subscribers and sends it to them in real time.









The national hub is therefore a single centralised pubsub service. There are however various architectures for implementing pubsub.

# 3.0 PubSub Architecture

Following Shen (2010[[1]](#footnote-1)), the architectural aspects of pubsub can be divided into:

* Subscription model
* Routing
* Topology

## 3.1 Subscription Model

The subscription model defines how a subscriber expresses to the pubsub service the pattern of events produced by publishers that they are interested in. There are three common models.

### 3.1.1 Topic-based

In a topic-based pubsub service each published event is associated with a single topic. A topic is also sometimes referred to as a group, channel or subject. A subscriber can subscribe to all events associated with a particular topic.

Topic-based pubsub is the oldest form of subscription model and the simplest to implement. It is also the most constrained, as you can only subscribe to a coarse grained topic.

To lessen the constraints, some variants of topic-based pubsub allow a hierarchical structure of topics. A subscription to a topic will receive all events published to that topic and any topics that lie below it in the topic tree.

Another approach is the filtered-topic variant. Here a subscriber can subscribe to a topic, but also define a set of filter criteria. If an event published to the topic meets the filter criteria for the individual subscriber, then the event is used. Note however, this filtering is normally done by the receiver of the topic event, not by the pubsub engine itself.

### 3.1.2 Content-based

In a content-based pubsub service, a subscription language is provided to allow a subscriber to define filtering criteria based on the content of events. The subscription language will allow the combination of attribute names and attribute values using a set of operators into a filter expression. Any events published that match the filter expression will be delivered to the subscriber.

Note in contrast to filtered-topic, this functionality is done within the pubsub engine.

Note there are no constraints on the structure of the content, as long as the subscription language can be operationalised across it. Therefore in some situations the content could be simple unstructured text.

As the filter expression can be as complex and as sophisticated as needed (only constrained by the expressiveness of the subscription language), this subscription model offers the most flexible and fine-grained approach.

However it is also computationally the most complex and resource intensive to implement.

### 3.1.3 Type-based

In a type-based pubsub service, each event is treated as an object that is an instance of a specific type or class. The type will have a formal definition including attributes (properties) and possibly methods.

A subscriber can subscribe to a particular type (similar to a topic) and also define a filter expression based on the type attribute values (similar to filtered-topic and content-based), where the filtering is done within the pubsub engine.

This approach lies between the simple topic-based approach and the complex content-based approach.

## 3.2 Routing

Once a published event has been matched to a subscription, it needs to be dispatched or routed through to the subscriber. The general case is considered here, where there maybe multiple independent pubsub nodes cooperating within a domain or federation. There are two common approaches.

### 3.2.1 Filter-based

In a filtered-based approach, events are routed between pubsub nodes based on content-based filtering at each successive node. The filtering is based on knowledge of remote subscriptions on other nodes. So a pubsub node may receive an event either directly from a publisher or forwarded to it by another pubsub node. If it holds local subscriptions that match the event it will service these. It will also check through filtering if any other pubsub nodes are holding subscriptions that would require the event, if so it will forward the event on.

The filter-based approach can be very efficient. However it imposes considerable complexity on pubsub nodes, as they now need to manage knowledge about subscriptions held on other nodes. The management of this information, the checking of the need to forward events on and the actual forwarding of events can produce considerable resource overhead.

### 3.2.2 Multicast-based

In a multicast-based approach, a multicast group is determined by a publisher using a set of shared rules for each event it wishes to publish. It multicasts the event to the group. Subscribers receive the event if they have joined the group.

## 3.3 Topology

The topology or organisation of the systems that make up a pubsub service can be classified as centralised, distributed or federated. In this context a distinction is made between physical nodes and service nodes. A service node represents a named autonomous pubsub service. That service node may be built from several physical nodes that are for example clustered to provide availability and scalability. In this discussion topology refers to the organisation of the service nodes.

### 3.3.1 Centralised

In a centralised topology there is a single service node which stores and manages all subscriptions, receives all event publications and runs an event dispatcher to forward on events to the relevant subscribers.

A Centralised topology provides a computationally efficient solution, as there are no overheads associated with routing between service nodes. However there are challenges of scalability as a single central service node needs to be capable of handling all pubsub traffic.

### 3.3.2 Distributed

In a distributed topology there are multiple pubsub service nodes that work in a co-operative manner.

A distributed topology overcomes the challenge of scalability but introduces new complexity in managing the matching of published events to subscriptions (this has already been outlined in the filter-based approach to routing).

There are two common approaches to implementing a distributed topology.

#### 3.3.2.1 Broker-based

A broker-based approach treats each pubsub service node as a member of a trusted domain. Events are considered to belong to a global content space. Segments of that content space are assigned to each service node through a consensus algorithm. Each service node has responsibility to manage subscriptions related to their assigned segment and dispatch associated events. Other service nodes will forward on events or subscriptions to the appropriate service node.

A broker-based approach is similar to database partitioning.

As with other broker-based approaches to distribution there is overhead with managing exchanging state information. In the case of a complete failure of a service node, then its responsibilities need to be redistributed across the remaining domain members dynamically.

Each service node essentially needs to be running the same pubsub system.

Although publishers will publish events to a named service node (their local pubsub service) and subscribers will manage their subscriptions through a named service node (again their local pubsub service), subscribers may receive events from any of the service nodes in the domain (dependant on how the content space has been segmented).

Finally, broker-based is complex to implement for content-based or type-based subscription models.

#### 3.3.2.2 Distributed Hash Table

A Distributed Hash Table (DHT) approach is similar to the broker-based approach in that the content space is segmented and assigned to distinct service nodes. However the process to do this is based on a consistent hash function that is applied to objects (published events, subscriptions and nodes) based on an associated identifier or key for each object. An object is assigned an owner node; this is done by hashing the key of the object and matching it to the closest node identifier.

Although the segmenting/partitioning process is simpler for DHT than the broker-based approach, all the disadvantages of the latter remain. In particular it is extremely complex to apply DHT to content-based or type-based subscription models.

### 3.3.3 Federated

In a federated typology there is a group of pubsub service domains that are provisioned and managed independently of each other. Each domain might be implemented as a centralised or distributed typology internally. Each domain might use completely different systems or product sets.

In a federated typology each domain must co-operate to exchange published events and subscriptions using appropriate routing (as for example outlined in the filter-based routing approach). This provides the major issue with the federated typology approach, all domains must use the same routing approach and there are no specific recognised industry standards that can be adopted to do this.

A second issue as already outlined for filter-based routing, is the potentially high overhead imposed on each domain to support the routing. In addition each domain must guarantee a Quality of Service and high availability for the federation to work.

# 4.0 Client Interaction

A client can interact with a pubsub service as either as publisher, subscriber or both.

## 4.1 Publisher

As a publisher a client will send event publications to the pubsub service. This is normally done under a push model; the client pushes the event to the pubsub service. This allows the client to have complete control over the timing of the publication process. The push might be over a synchronous mechanism (such as a RESTful or SOAP web service) or could be an asynchronous mechanism (using a messaging protocol such as AMQP or MQTT, or a propriety service such as NHS MESH).

## 4.2 Subscriber

As a subscriber a client will receive published events that match subscriptions they have created on the pubsub service. The pubsub service can use two delivery models.

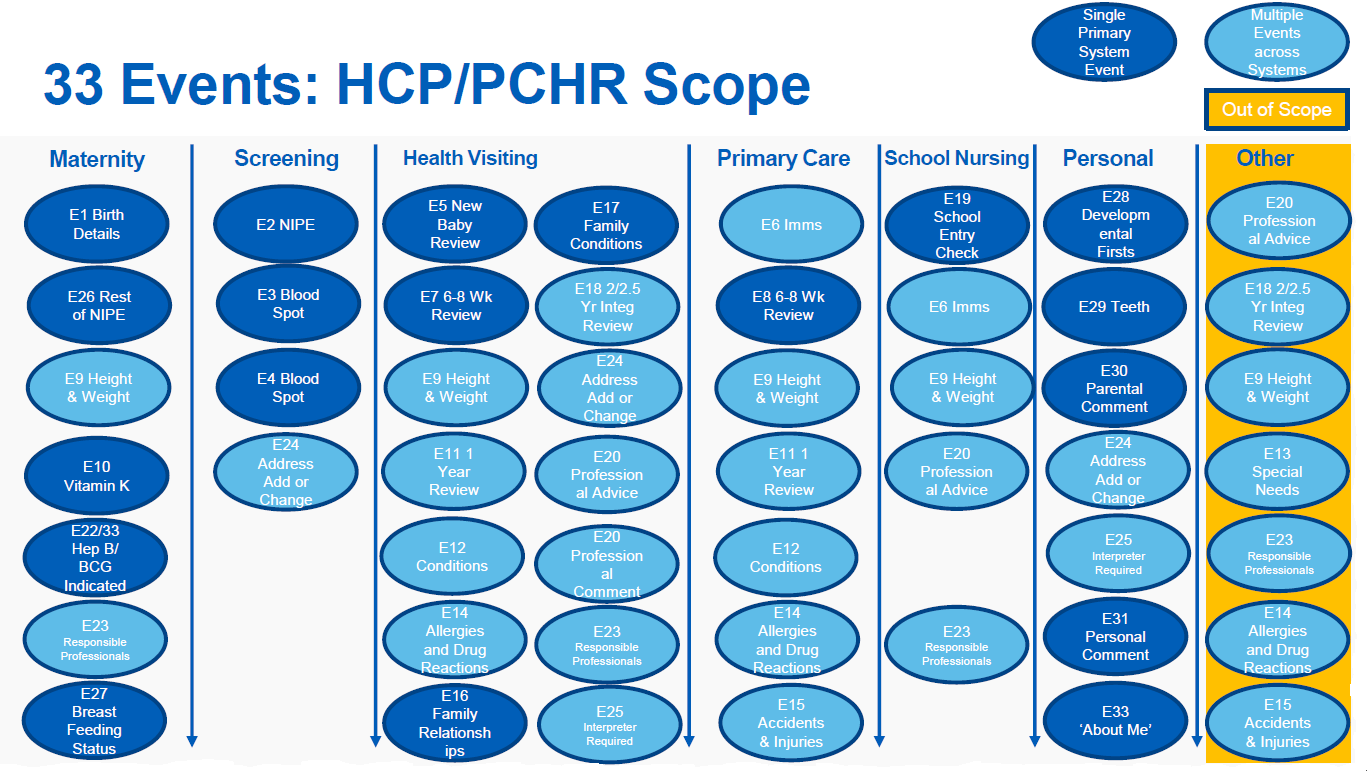
The first is a push model; the pubsub service will push an event to the subscriber. To do this, the subscriber must tell the pubsub service the client endpoint on which it will be listening for events. The advantage of the push model is that a client will receive events as soon as the pubsub can deliver them. The disadvantage is that a client must implement logic to listen for incoming events (which can be non-trivial). Also the pubsub service must decide what to do if it cannot connect to the client – what is the retry policy? What does it do if the client is unavailable for a long period of time?

The second is the pull or polling model, the client asks (or polls) the pubsub service for events. The advantage of the polling model is that control of the process lies with the client; it can decide when to check for events. The disadvantage is that if a client wants a near real-time feed of events it will need to implement a very short polling interval, which will impose a significant processing burden on the pubsub service if there are a large number of clients doing the same.

As for a publisher, both synchronous and asynchronous protocols can be used for push or pull models.

# 5.0 Child Health PubSub Considerations

For Child Health an initial 33 events have been defined to be published.



The information content of these events will be modelled using Fast Healthcare Interoperability Resources (FHIR).

These can be published by a variety of providers, including:

* Maternity
* National Patient Demographics Service (PDS)
* National Screening Services
* Primary Care
* Health Visiting
* School Nursing
* Personal Health Record (ePCHR)
* Child Health Record Departments (CHRD)

These can be subscribed to by:

* Primary Care
* Health Visiting
* School Nursing
* Personal Health Record (ePCHR)
* Child Health Record Departments (CHRD)
* Public Health England

## 5.1 Subscription Model

For many of the subscribers they will be interested in events relating to a specific individual. In this use case a simple topic-based subscription model would be sufficient, where the topic might be the individual’s NHS number. However this would become onerous where a single subscriber is interested in a large number of individuals. For example a primary care provider may have many hundreds of children registered with them. Using a topic-based subscription they would have to create an individual subscription for each child. An alternative would be take out a single subscription for all children registered with their GP practise. This would require a content-based subscription model.

Some subscribers may have an interest in a large number of individuals (often called a cohort), where the membership of the cohort changes relatively often. With a topic-based subscription model, management of the individual subscriptions (creating and deleting) would become onerous. A content-based subscription model would be more effective.

Some subscribers (such as Public Health England) are not interested in events relating to individuals, but just what the event is about (for example immunisations). A topic-based subscription model could be used where the event is published to a topic relating to the event type. However this precludes using a topic to identify an individual. One way round this is to publish the same event to multiple topics. This however produces duplication and extra overhead. It can easily become unmanageable when the classes of topics expands and a single event for example is relevant to a large group of topics. It also imposes additional burden on the publisher, which now has to implement potentially complex topic classification logic. Again a content-based subscription model would be more effective.

As stated above events will be modelled using FHIR standards. This also provides a set of standard exchange frameworks which also includes a subscription mechanism:

The subscription resource is used to define a push based subscription from a server to another system. Once a subscription is registered with the server, the server checks every resource that is created or updated, and if the resource matches the given criteria, it sends a message on the defined "channel" so that another system is able to take an appropriate action. The server is able to send notifications without any information about the matching resource, or with the entire resource.

Several different types of channels are supported:

* **rest-hook**: A post is made to the URL. If the subscription requests that the whole resource is included, the URL is interpreted as the service base
* **websocket**: An PING message is sent to the designated URI
* **email/sms**: A notification is send to nominated email address or SMS number
* **message**: The resource is sent to the application identified in the URI as a [message](https://www.hl7.org/fhir/messaging.html)

The criteria referred to in the extract from the standard is a generalised search expression over FHIR resources. As such it equates to a content-based / type-based subscription model.

Based on these considerations it seems that a simple topic-based subscription model will not be adequate for Child Health. A type-based or content-based subscription model seems more suitable.

However practical exploration of the suitability and viability of the different subscription models is one of the key objectives of the alpha or proof of concept phase of the Events Management Service. This will involve collaboration with suppliers of systems that will act as both publishers and subscribers.

## 5.2 Routing

Of the two routing approaches, multicast is the most problematic as it requires appropriate network configuration across all networks. As the health and care sectors use an inter-network or networks, each managed by a separate organisation, implementing the appropriate network rules and testing across the whole sector would be very difficult.

Therefore filter-based routing seems the most appropriate choice when the typology requires routing.

## 5.3 Topology

The key requirement of the distributed typology option is the need for all service nodes to work within a single trusted domain. Where this domain is equated with a single organisation this seems feasible. However positing a distributed typology option across autonomous entities such as individual health providers or local health economy Integrated Digital Care Records (IDCR) is not practical due to the technical constraints (as outlined previously). Therefore use of a distributed typology across autonomous entities is not deemed suitable.

A centralised typology provides the simplest approach, negating the need for routing. It however presents challenges of scalability and availability. It is the preferred approach for Child Health as it has been explicitly stated in the NHS England Children’s Health Digital Strategy.

However this presents some policy dissonance, as there is also NHS England initiatives to establish autonomous local IDCRs. Such IDCRs are expected to operate within a wider federated model. It could therefore be argued each IDCR could operate its own pubsub service node and use a federated typology.

As stated previously, the major issue with a federated typology is the lack of specific recognised industry standards for federated pubsub. There should be no confusion here around the use of the terms federation and standards. When talking about the federation of IDCRs there seems to be no clear conceptual or logical definition of this within the health and social care sectors (there isn’t even an agreed definition of what an IDCR is). There is often a conflation of the concept of federation with a set of standards that Integrating the Healthcare Enterprise (IHE) profile, namely XDS and XCA.

Firstly these are profiles of existing standards published by other organisations, IHE does not create standards.

Secondly XDS and XCA relate to managing documents in a registry and repository, and allowing access to documents (query retrieve) from a remote registry/repository. The actual standard XDS is based on is the OASIS ebXML Registry Services and Protocols Version 3.0. This defines its own pubsub mechanism called Event Notification:

Event Notification feature allows OASIS ebXML Registries to notify its users and / or other registries about events of interest. It allows users to stay informed about registry events without being forced to periodically poll the registry. It also allows a registry to propagate internal changes to other registries whose content might be affected by those changes.

ebXML registries support content-based Notification where interested parties express their interest in form of a query. This is different from subject–based (sometimes referred to as topic-based) notification, where information is categorized by subjects and interested parties express their interests in those predefined subjects.

The standard also defines Cooperating Registries Support (aka federation). This is mainly focused on federated queries and has no explicit support for federated pubsub (though no doubt some sort of mashup could probably be made from the standard).

Thirdly the standard has not been mandated by NHS England for all IDCRs, therefore there will be existing and future IDCRs that would require significant re-engineering or re-platforming to support it.

There is also the deployment issue that there is currently limited IDCR coverage for the whole of England. If it is assumed that each Local Sustainability and Transformation Plan (STP) would equate with a local IDCR, there would need to be 44 IDCRs to cover England to provide a viable federation approach (irrespective of how this would be achieved). However there is no visible national overview of commitments, plans or roadmaps to achieve this.

Although the centralised typology is the preferred approach, practical exploration of and experimental design of a federated pubsub protocol will be undertaken as part of the alpha or proof of concept phase of the Events Management Service.

## 5.4 Client Interaction

With regard to client interaction for both publisher and subscriber, child health has to be pragmatic, and accommodate a wide range of end point systems with a wide range of communication capabilities. Therefore multi-channel support is needed (async, sync, push, pull etc.) One important consideration is channel support must be network agnostic. Some endpoints (particularly with regard to Personal Health Record systems for child) are hosted on the internet rather than N3.

1. Shen, Haiying. "Content-based publish/subscribe systems." *Handbook of Peer-to-Peer Networking*. Springer US, 2010. 1333-1366. [↑](#footnote-ref-1)