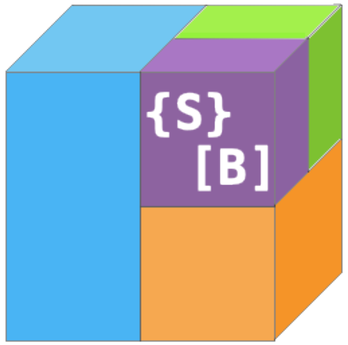


{S}[B] SchemaBlocks

GA4GH Standards Documentation and Alignment Initiative

Global Alliance
for Genomics & Health
Collaborate. Innovate. Accelerate.

Global Alliance
for Genomics & Health

Collaborate. Innovate. Accelerate.

Scientists Seek Order to Potential Confusion of Gene Data
Bloomberg - Drew Armstrong & Robert Langreth
June 5, 2013

Q&A: David Altshuler on How to Share Millions of Human Genomes
Science - Jocelyn Kaiser
June 7, 2013

DNA data to be shared worldwide in medical research project
The Guardian - Ian Sample
June 5, 2013

Geneticists push for global data-sharing
Nature - Erika Check Hayden
June 5, 2013

Accord Aims to Create Global Trove of Genetic Data
The New York Times - Gina Kolata
June 5, 2013

Global Alliance for Genomics & Health

Poking Holes in Genetic Privacy
The New York Times - Gina Kolata
June 16, 2013

New alliance aims to create international system for sharing genomic data
The Globe and Mail - By André Picard
June 5, 2013

White House Open Science 'Champions' Highlights Genomic Data Pioneers
GenomeWeb
June 19, 2013

Our Genes, Their Secrets
The New York Times
June 18, 2013

Une alliance pour partager les données génomiques et cliniques
Le Monde - Sandrine Cabut
June 14, 2013

Global alliance to create framework for sharing genomic data
The Boston Globe - Carolyn Y. Johnson
June 5, 2013

GENOMICS

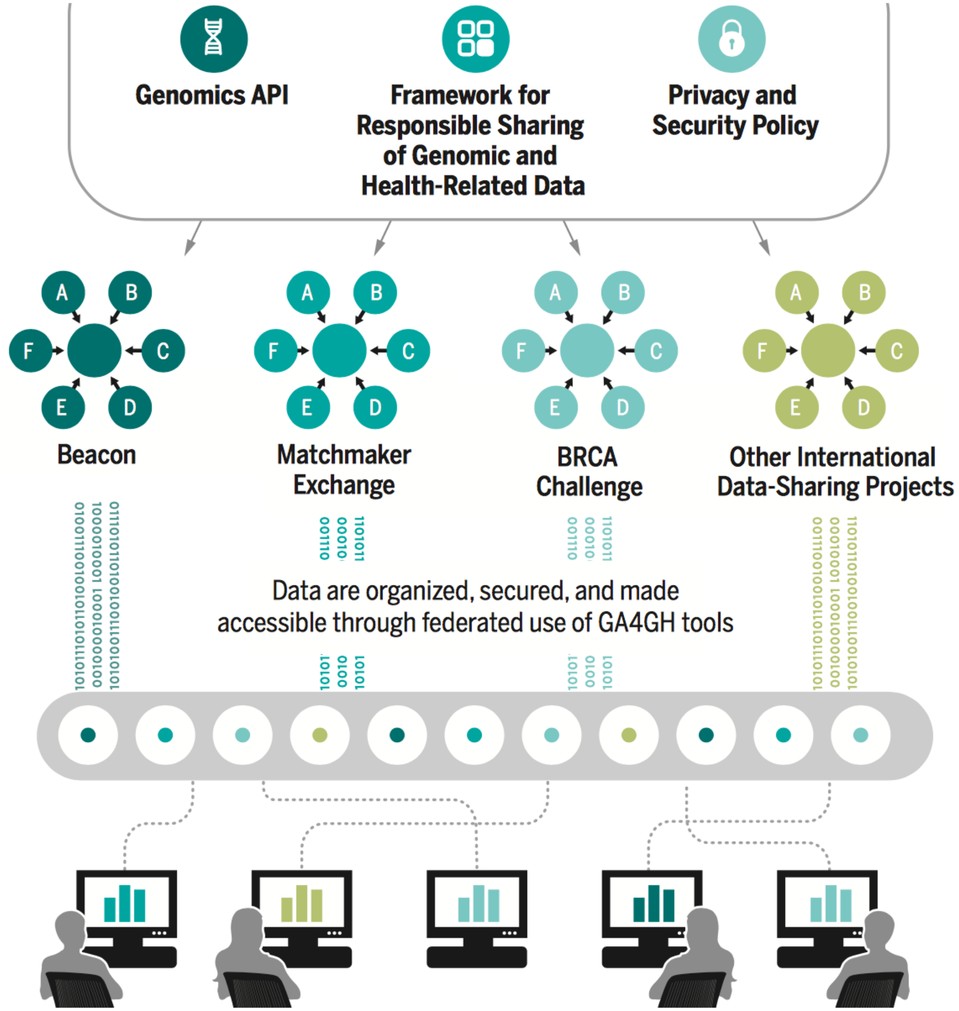
A federated ecosystem for sharing genomic, clinical data

Silos of genome data collection are being transformed into seamlessly connected, independent systems

The Global Alliance for Genomics and Health*

SCIENCE 10 JUNE 2016 • VOL 352 ISSUE 6291

A federated data ecosystem. To share genomic data globally, this approach furthers medical research without requiring compatible data sets or compromising patient identity.

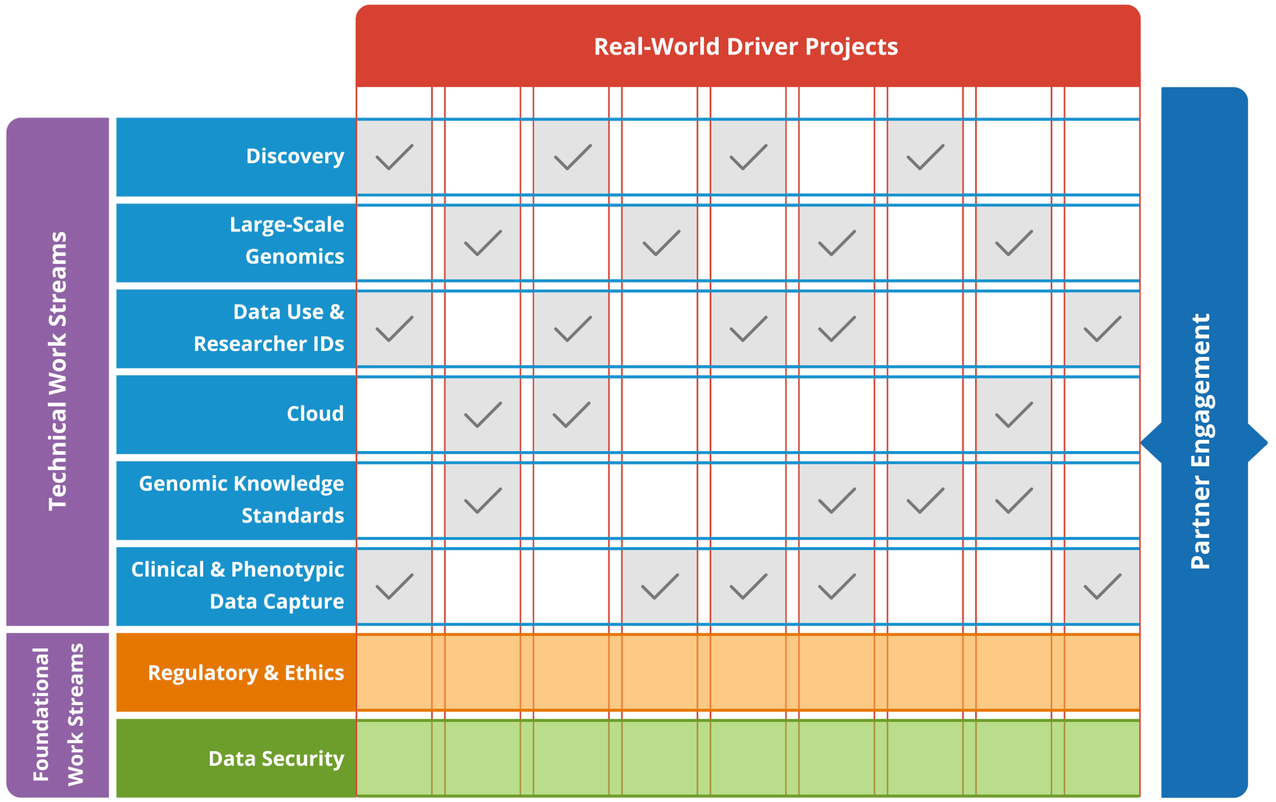


Global Alliance for Genomics & Health

Organizational Structure - Work Streams & Driver Projects

GA4GH Driver Projects **are real-world** genomic **data** initiatives that help guide our development efforts and pilot our tools. Stakeholders around the globe advocate, mandate, implement, and use our **frameworks** and standards in their local contexts.

GA4GH Foundational and Technical Work Streams develop standards and tools that are designed to overcome technical and regulatory hurdles to international genomic data-sharing.



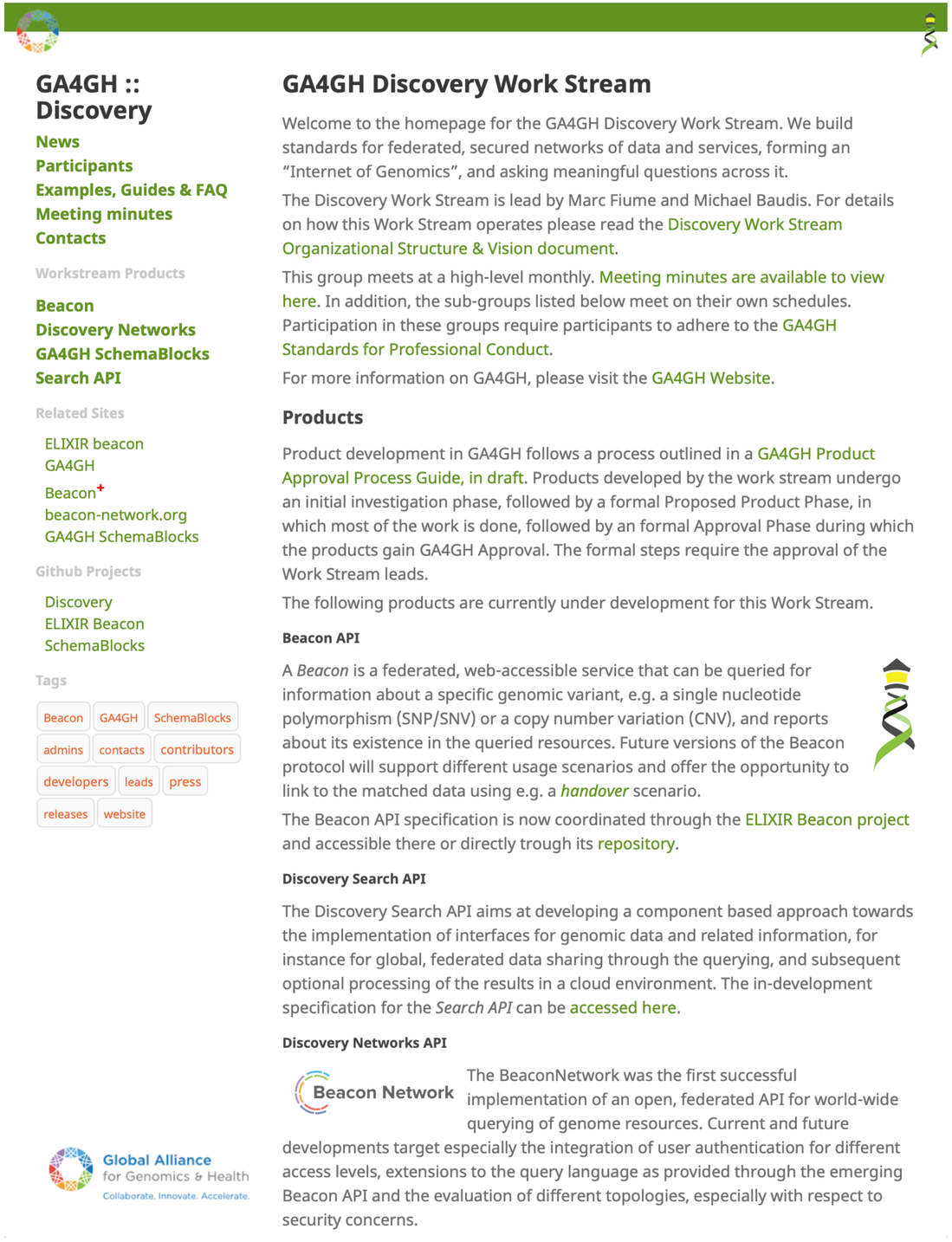
The GA4GH Partner Engagement initiative facilitates two-way dialogue with the international community, including national initiatives, major health care centres, and patient advocacy groups.

GA4GH :: Discovery

A Work Stream of The Global Alliance for Genomics and Health

We build standards for federated, secured networks of data and services, forming an "Internet of Genomics", and asking meaningful questions across it.

- Marc Fiume
  - Discovery Networks
  - Search API / Data Discovery
- Michael Baudis
  - Beacon
  - SchemaBlocks **{S}[B]**



GA4GH :: Discovery

News
Participants
Examples, Guides & FAQ
Meeting minutes
Contacts

Workstream Products

Beacon
Discovery Networks
GA4GH SchemaBlocks
Search API

Related Sites

ELIXIR beacon
GA4GH
Beacon+
beacon-network.org
GA4GH SchemaBlocks

Github Projects

Discovery
ELIXIR Beacon
SchemaBlocks

Tags

Beacon GA4GH SchemaBlocks admins contacts contributors developers leads press releases website

Global Alliance for Genomics & Health
Collaborate. Innovate. Accelerate.

GA4GH Discovery Work Stream

Welcome to the homepage for the GA4GH Discovery Work Stream. We build standards for federated, secured networks of data and services, forming an "Internet of Genomics", and asking meaningful questions across it.

The Discovery Work Stream is lead by Marc Fiume and Michael Baudis. For details on how this Work Stream operates please read the Discovery Work Stream Organizational Structure & Vision document.

This group meets at a high-level monthly. Meeting minutes are available to view here. In addition, the sub-groups listed below meet on their own schedules. Participation in these groups require participants to adhere to the GA4GH Standards for Professional Conduct.

For more information on GA4GH, please visit the GA4GH Website.

Products

Product development in GA4GH follows a process outlined in a GA4GH Product Approval Process Guide, in draft. Products developed by the work stream undergo an initial investigation phase, followed by a formal Proposed Product Phase, in which most of the work is done, followed by an formal Approval Phase during which the products gain GA4GH Approval. The formal steps require the approval of the Work Stream leads.

The following products are currently under development for this Work Stream.

Beacon API

A *Beacon* is a federated, web-accessible service that can be queried for information about a specific genomic variant, e.g. a single nucleotide polymorphism (SNP/SNV) or a copy number variation (CNV), and reports about its existence in the queried resources. Future versions of the Beacon protocol will support different usage scenarios and offer the opportunity to link to the matched data using e.g. a *handover* scenario.

The Beacon API specification is now coordinated through the ELIXIR Beacon project and accessible there or directly trough its repository.

Discovery Search API

The Discovery Search API aims at developing a component based approach towards the implementation of interfaces for genomic data and related information, for instance for global, federated data sharing through the querying, and subsequent optional processing of the results in a cloud environment. The in-development specification for the *Search API* can be accessed here.

Discovery Networks API

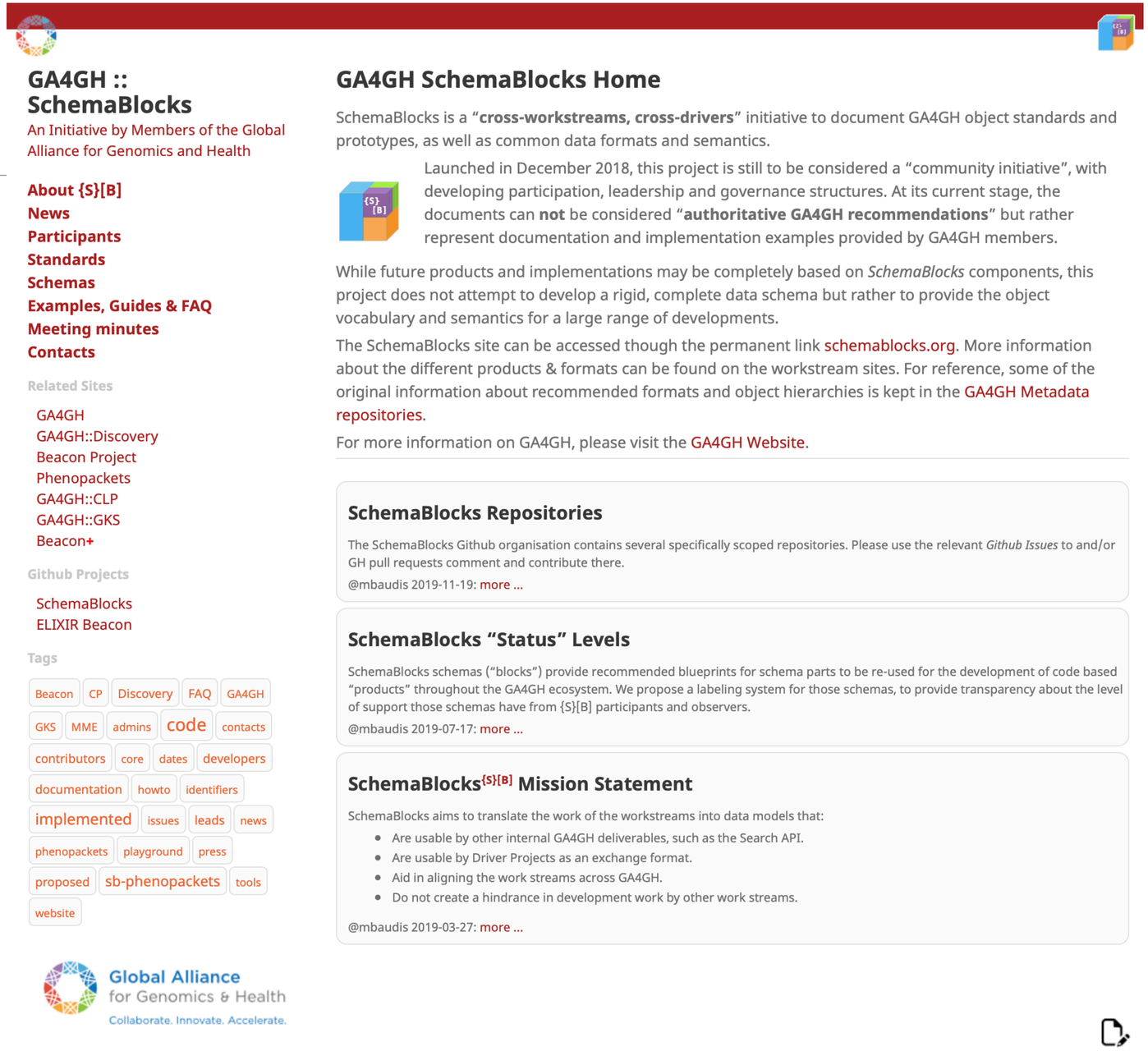
Beacon Network

The BeaconNetwork was the first successful implementation of an open, federated API for world-wide querying of genome resources. Current and future developments target especially the integration of user authentication for different access levels, extensions to the query language as provided through the emerging Beacon API and the evaluation of different topologies, especially with respect to security concerns.

GA4GH {S}[B]
SchemaBlocks

- “cross-workstreams, cross-drivers” initiative to document GA4GH object standards and prototypes, data formats and semantics
- launched in December 2018
- documentation and implementation examples provided by GA4GH members
- no attempt to develop a rigid, complete data schema
- object vocabulary and semantics for a large range of developments
- currently not “authoritative GA4GH recommendations”
- recognized in GA4GH roadmap as element in "TASC" effort

schemablocks.org



GA4GH :: SchemaBlocks

An Initiative by Members of the Global Alliance for Genomics and Health

About {S}[B]
News
Participants
Standards
Schemas
Examples, Guides & FAQ
Meeting minutes
Contacts

Related Sites

GA4GH
GA4GH::Discovery
Beacon Project
Phenopackets
GA4GH::CLP
GA4GH::GKS
Beacon+

Github Projects

SchemaBlocks
ELIXIR Beacon

Tags

Beacon CP Discovery FAQ GA4GH GKS MME admins code contacts contributors core dates developers documentation howto identifiers implemented issues leads news phenopackets playground press proposed sb-phenopackets tools website

Global Alliance for Genomics & Health
Collaborate. Innovate. Accelerate.

GA4GH SchemaBlocks Home

SchemaBlocks is a “**cross-workstreams, cross-drivers**” initiative to document GA4GH object standards and prototypes, as well as common data formats and semantics.

Launched in December 2018, this project is still to be considered a “community initiative”, with developing participation, leadership and governance structures. At its current stage, the documents can **not** be considered “**authoritative GA4GH recommendations**” but rather represent documentation and implementation examples provided by GA4GH members.

While future products and implementations may be completely based on *SchemaBlocks* components, this project does not attempt to develop a rigid, complete data schema but rather to provide the object vocabulary and semantics for a large range of developments.

The SchemaBlocks site can be accessed though the permanent link schemablocks.org. More information about the different products & formats can be found on the workstream sites. For reference, some of the original information about recommended formats and object hierarchies is kept in the GA4GH Metadata repositories.

For more information on GA4GH, please visit the GA4GH Website.

SchemaBlocks Repositories

The SchemaBlocks Github organisation contains several specifically scoped repositories. Please use the relevant *Github Issues* to and/or GH pull requests comment and contribute there.

@mbaudis 2019-11-19: more ...

SchemaBlocks “Status” Levels

SchemaBlocks schemas (“blocks”) provide recommended blueprints for schema parts to be re-used for the development of code based “products” throughout the GA4GH ecosystem. We propose a labeling system for those schemas, to provide transparency about the level of support those schemas have from {S}[B] participants and observers.

@mbaudis 2019-07-17: more ...

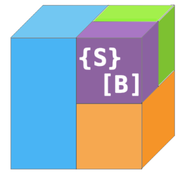
SchemaBlocks{S}[B] Mission Statement

SchemaBlocks aims to translate the work of the workstreams into data models that:

- Are usable by other internal GA4GH deliverables, such as the Search API.
- Are usable by Driver Projects as an exchange format.
- Aid in aligning the work streams across GA4GH.
- Do not create a hindrance in development work by other work streams.

@mbaudis 2019-03-27: more ...

SchemaBlocks - A GA4GH Community Initiative



GA4GH SchemaBlocks Home

SchemaBlocks is a **"cross-workstreams, cross-drivers"** initiative to document GA4GH object standards and prototypes, as well as common data formats and semantics.

Launched in December 2018, this project is still to be considered a "community initiative", with developing participation, leadership and governance structures. At its current stage, the documents can **not** be considered "**authoritative GA4GH recommendations**" but rather represent documentation and implementation examples provided by GA4GH members.

While future products and implementations may be completely based on *SchemaBlocks* components, this project does not attempt to develop a rigid, complete data schema but rather to provide the object vocabulary and semantics for a large range of developments.

The SchemaBlocks site can be accessed though the permanent link schemablocks.org. More information about the different products & formats can be found on the workstream sites. For reference, some of the original information about recommended formats and object hierarchies is kept in the GA4GH Metadata repositories.

For more information on GA4GH, please visit the GA4GH Website.

SchemaBlocks "Status" Levels

SchemaBlocks schemas ("blocks") provide recommended blueprints for schema parts to be re-used for the development of code based "products" throughout the GA4GH ecosystem. We propose a labeling system for those schemas, to provide transparency about the level of support those schemas have from {S}[B] participants and observers.

Proposed {S}[B] Status Levels

The current status level of thiose recommendations is "proposed".

- `playground`
  - early development or import stage, of any quality
  - no recommendation; existence does not mean any current or future {S}[B] support
- `proposed`
  - at least some {S}[B] contributors are in favour of such a block
  - the code may undergo considerable maturation
  - not recommended for integration into products w/o close tracking
  - contributions and discussions are encouraged
- `implemented`
  - mature block which is implemented in one or more {S}[B] aligned schemas
  - may be extended from a core block or be too specific for general ("core") usability
- `core`
  - a schema block with recommended use
  - stable through minor version changes
  - has to be used in at least 2 standards/products approved by the GA4GH Steering Committee

SchemaBlocks{S}[B] Mission Statement

SchemaBlocks aims to translate the work of the workstreams into data models that:

- Are usable by other internal GA4GH deliverables, such as the Search API.
- Are usable by Driver Projects as an exchange format.
- Aid in aligning the work streams across GA4GH.
- Do not create a hindrance in development work by other work streams.

After discussions with stakeholders from GA4GH work streams and driver projects who create data models (such as Phenopackets, Search API) or who would use SchemaBlocks for the development of their APIs and data exchange formats (Beacon, EGA, GeL), the SchemaBlocks team has come up with the following principles for this initiative:

Work Stream Interactions

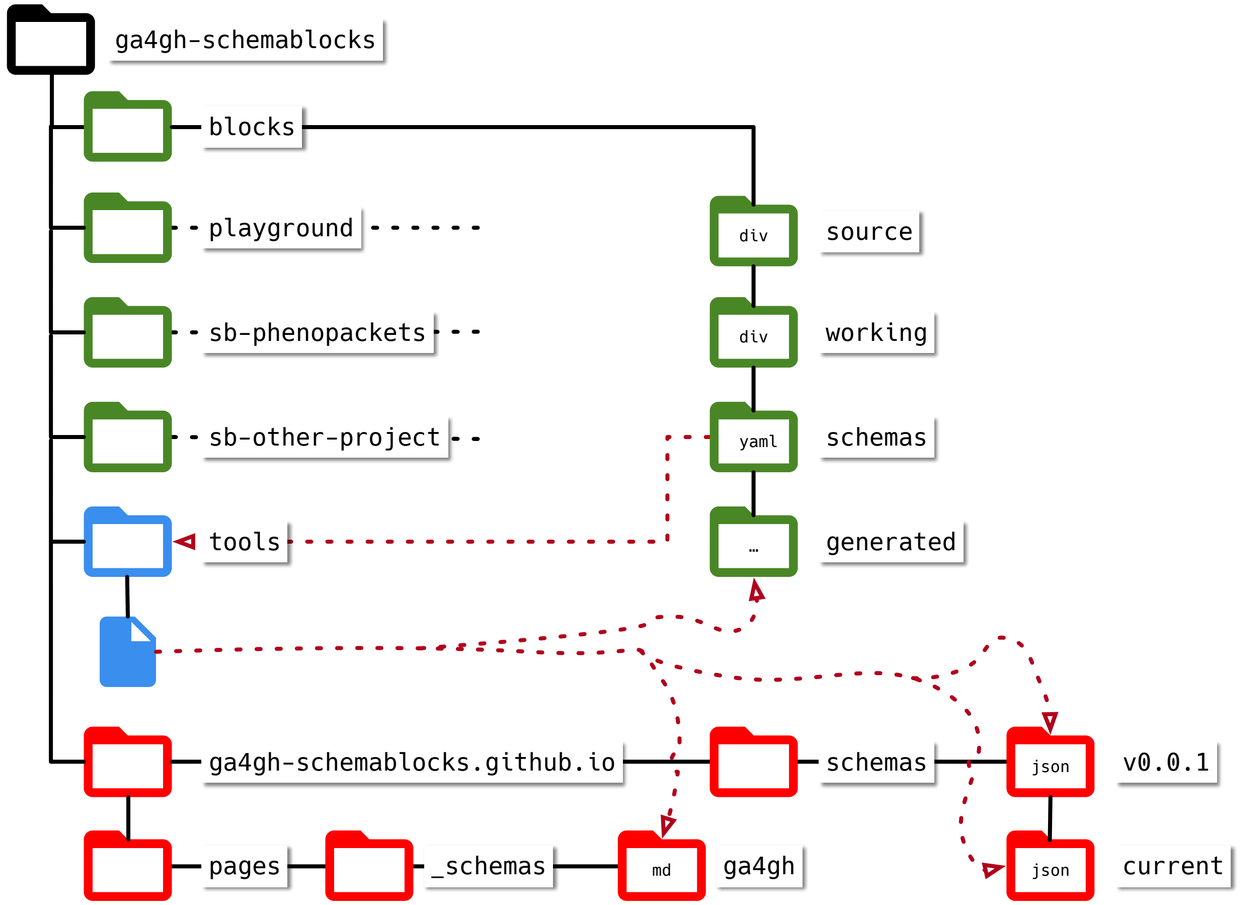
Work streams will continue to create standards proposals and their own coherent project implementations, but will work with the SchemaBlocks group to write the Blocks that will come from their own work and are considered of overarching use. Generally, primary work stream and driver project outputs will live in their own spaces outside of SchemaBlocks, with shareable, mature elements - code, documentation, implementation snapshots - being represented in {S}[B].

{S}[B] SchemaBlocks Github Repository Structure

blocks repositories

conversion/validation tools

website repository
(Markdown w/ YAML for Github Pages)



Dissection & Transformation

```
...
// See http://build.fhir.org/datatypes and http://build.fhir.org/condition-definitions.html#Condition.onset_x_
// In FHIR this is represented as a UCUM measurement - http://unitsofmeasure.org/trac/
message Age {

    // The :ref:`ISO 8601<metadata_date_time>` age of this object as ISO8601
    // duration or time intervals. The use of time intervals makes an additional
    // anchor unnecessary (i.e. DOB and age can be represented as start-anchored
    // time interval, e.g. 1967-11-21/P40Y10M05D)
    string age = 1;
}

message AgeRange {
    Age start = 1;
    Age end = 2;
}

// Message to indicate a disease (diagnosis) and its recorded onset.
message Disease {
    // The identifier of this disease e.g. MONDO:0007043, OMIM:101600, Orphanet:710, DOID:14705 (note these are all equivalent)
    OntologyClass term = 1;

    // The onset of the disease. The values of this will come from the HPO onset hierarchy
    // i.e. subclasses of HP:0003674
    // FHIR mapping: Condition.onset
    oneof onset {
        Age age_of_onset = 2;
        AgeRange age_range_of_onset = 3;
        OntologyClass class_of_onset = 4;
    }

    // Disease staging, the extent to which a disease has developed.
    // For cancers, see https://www.cancer.gov/about-cancer/diagnosis-staging/staging
    // Valid values include child terms of NCIT:C28108 (Disease Stage Qualifier)
    repeated OntologyClass disease_stage = 5;
...
```

- Excerpt from Phenopackets v1.0 Schema
- written in *Protocol Buffers* (Google's data serializing format)
- separate documentation rendered in "ReadTheDocs"

Use Case Transforming Phenopackets objects (here "Age") into JSON Schema documents with (proposed) stable id and address as well as "human readable" documentation & examples.

Dissection & Transformation

```
...
// See http://build.fhir.org/datatypes and http://build.fhir.org/condition-definitions.html#Condition.onset_x_
// In FHIR this is represented as a UCUM measurement - http://unitsofmeasure.org/trac/
message Age {

    // The :ref:`ISO 8601<metadata_date_time>` age of this object as ISO8601
    // duration or time intervals. The use of time intervals makes an additional
    // anchor unnecessary (i.e. DOB and age can be represented as start-anchored
    // time interval, e.g. 1967-11-21/P40Y10M05D)
    string age = 1;
}

message AgeRange {
    Age start = 1;
    Age end = 2;
}

// Message to indicate a disease (diagnosis) and its recorded onset.
message Disease {
    // The identifier of this disease e.g. MONDO:0007043, OMIM:101600, Orphanet:710, DOID:14705 (note these are all equivalent)
    OntologyClass term = 1;

    // The onset of the disease. The values of this will come from the HPO onset hierarchy
    // i.e. subclasses of HP:0003674
    // FHIR mapping: Condition.onset
    oneof onset {
        Age age_of_onset = 2;
        AgeRange age_range_of_onset = 3;
        OntologyClass class_of_onset = 4;
    }

    // Disease staging, the extent to which a disease has developed.
    // For cancers, see https://www.cancer.gov/about-cancer/diagnosis-staging/staging
    // Valid values include child terms of NCIT:C28108 (Disease Stage Qualifier)
    repeated OntologyClass disease_stage = 5;
...
```

32 lines (31 sloc) | 872 Bytes — Raw | Blame | History

```yaml
"$schema": http://json-schema.org/draft-07/schema#
"$id": https://schemablocks.org/schemas/sb-phenopackets/Age/v0.0.1
title: Age
description: Age
type: object
meta:
  contributors:
    - description: "Michael Baudis"
      id: "orcid:0000-0002-9903-4248"
    - description: "Jules Jacobsen"
      id: "orcid:0000-0002-3265-15918"
    - description: "Peter Robinson"
      id: "orcid:0000-0002-0736-91998"
  provenance:
    - description: Phenopackets
      id: 'https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst'
  used_by:
    - description: Phenopackets
      id: 'https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst'
  sb_status: implemented
properties:
  age:
    type: string
    description: Age as ISO8601 period
    examples:
      - 'P12Y'

required:
  - age
additionalProperties: false
examples:
  - age: 'P14Y'
```

- Separate {S}[B] repository for parental project
- here "sb-phenopackets"
- individual schema documents for each original object
- (currently) manual re-write into JSON Schema documents (YAML version), including metadata header (id, provenance ...)
- versioned

Dissection & Transformation

```
...
// See http://build.fhir.org/datatypes and http://build.fhir.org/condition-definitions.html#Condition.onset_x_
// In FHIR this is represented as a UCUM measurement - http://unitsofmeasure.org/trac/
message Age {

    // The :ref:`ISO 8601<metadata_date_time>` age of this object as ISO8601
    // duration or time intervals. The use of time intervals makes an additional
    // anchor unnecessary (i.e. DOB and age can be represented as start-anchored
    // time interval, e.g. 1967-11-21/P40Y10M05D)
    string age = 1;
}

message AgeRange {
    Age start = 1;
    Age end = 2;
}

// Message to indicate a disease (diagnosis) and its recorded onset.
message Disease {
    // The identifier of this disease e.g. MONDO:0007043, OMIM:101600, Orphanet:710, DOID:14705 (note these are all equivalent)
    OntologyClass term = 1;

    // The onset of the disease. The values of this will come from the HPO onset hierarchy
    // i.e. subclasses of HP:0003674
    // FHIR mapping: Condition.onset
    oneof onset {
        Age age_of_onset = 2;
        AgeRange age_range_of_onset = 3;
        OntologyClass class_of_onset = 4;
    }

    // Disease staging, the extent to which a disease has developed.
    // For cancers, see https://www.cancer.gov/about-cancer/diagnosis-staging/staging
    // Valid values include child terms of NCIT:C28108 (Disease Stage Qualifier)
    repeated OntologyClass disease_stage = 5;
...
```

32 lines (31 sloc) | 872 Bytes | Raw | Blame | History

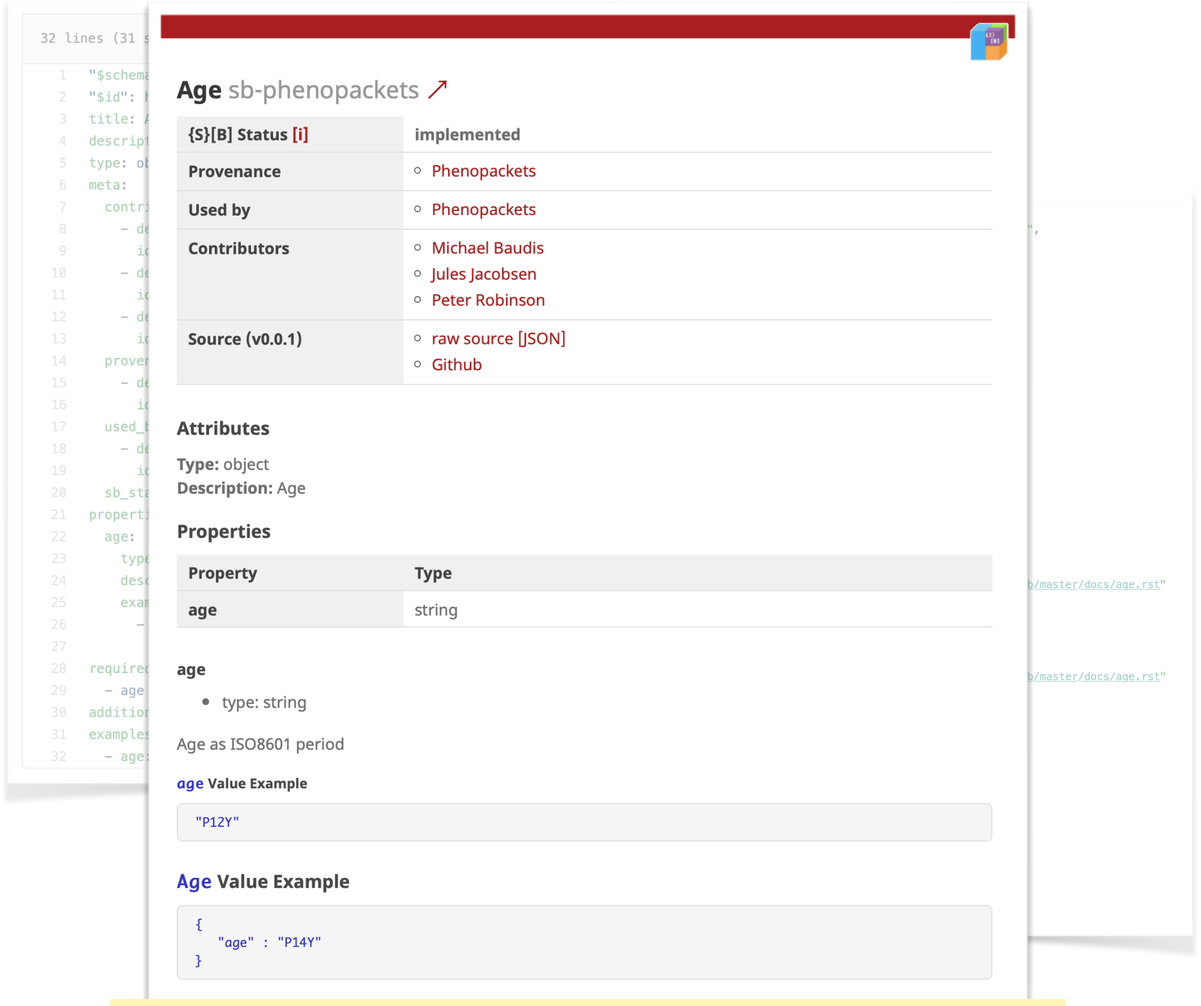
```
"$schema": http://json-schema.org/draft-07/sc
"$id": https://schemablocks.org/schemas/sb-ph
title: Age
description: Age
type: object
meta:
  contributors:
    - description: "Michael Baudis"
      id: "orcid:0000-0002-9903-4248"
    - description: "Jules Jacobsen"
      id: "orcid:0000-0002-3265-15918"
    - description: "Peter Robinson"
      id: "orcid:0000-0002-0736-91998"
  provenance:
    - description: Phenopackets
      id: 'https://github.com/phenopackets/ph
  used_by:
    - description: Phenopackets
      id: 'https://github.com/phenopackets/ph
  sb_status: implemented
properties:
  age:
    type: string
    description: Age as ISO8601 period
    examples:
      - 'P12Y'

required:
  - age
additionalProperties: false
examples:
  - age: 'P14Y'
```


```
{
  "$id": "https://schemablocks.org/schemas/sb-phenopackets/Age/v0.0.1",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "additionalProperties": "",
  "description": "Age",
  "examples": [
    {
      "age": "P14Y"
    }
  ],
  "meta": {
    "contributors": [
      {
        "description": "Michael Baudis",
        "id": "orcid:0000-0002-9903-4248"
      },
      {
        "description": "Jules Jacobsen",
        "id": "orcid:0000-0002-3265-15918"
      },
      {
        "description": "Peter Robinson",
        "id": "orcid:0000-0002-0736-91998"
      }
    ],
    "provenance": [
      {
        "description": "Phenopackets",
        "id": "https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst"
      }
    ],
    "sb_status": "implemented",
    "used_by": [
      {
        "description": "Phenopackets",
        "id": "https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst"
      }
    ]
  },
  "properties": {
    "age": {
      "description": "Age as ISO8601 period",
      "examples": [
        "P12Y"
      ],
      "type": "string"
    }
  },
  "required": [
    "age"
  ],
  "title": "Age",
  "type": "object"
}
```


- schema documents are programmatically converted into different outputs
- a versioned JSON document serves as canonical reference for integration into other products/schemas

Dissection & Transformation

```
...
// See http://build.fhir.org/datatypes and http://build.fhir.org/condition-definitions.html#Condition.onset_x_
// In FHIR this is represented as a UCUM measurement - http://unitsofmeasure.org/trac/
message Age {

    // The :ref:`ISO 8601<metadata_date_time>` age of this object as ISO8601
    // duration or time intervals. The use of time intervals makes an additional
    // anchor unnecessary (i.e. DOB and age can be represented as start-anchored
    // time interval, e.g. 1967-11-21/P40Y10M05D)
    string age = 1;
}

message AgeRange {
    Age start = 1;
    Age end = 2;
}

// Message to indicate a disease (diagnosis) and its recorded onset.
message Disease {
    // The identifier of this disease e.g. MONDO:0007043, OMIM:101600, Orphanet:710, DOID:14705 (note these are all equivalent)
    OntologyClass term = 1;

    // The onset of the disease. The values of this will come from the HPO onset hierarchy
    // i.e. subclasses of HP:0003674
    // FHIR mapping: Condition.onset
    oneof onset {
        Age age_of_onset = 2;
        AgeRange age_range_of_onset = 3;
        OntologyClass class_of_onset = 4;
    }

    // Disease staging, the extent to which a disease has developed.
    // For cancers, see https://www.cancer.gov/about-cancer/diagnosis-staging/staging
    // Valid values include child terms of NCIT:C28108 (Disease Stage Qualifier)
    repeated OntologyClass disease_stage = 5;
...
```



- schema documents are programmatically converted into different outputs
- a Markdown document with "Jekyll" header is auto-converted by Github into a complete website document, including inline code examples

{S}[B] SchemaBlocks ***JSON Schema*** document format

- {S}[B] "blocks" are written in the YAML version of a JSON Schema document format
  - convenience choice - flexibility, readability, tooling ...
  - ***not*** implying specific semantics beyond some format conventions - extensible for use-case driven requirements
- the meta part (itself defined as a schema "block") contains housekeeping information
  - reference address & version
  - provenance & use cases
  - sb_status about "blessing level"
- the properties part defines the attributes including their description and usage examples
  - descriptions & examples provide the core documentation which is deparsed t0 the website documents


```
"$schema": http://json-schema.org/draft-07/schema#
"$id": https://schemablocks.org/schemas/ga4gh/AgeRange/v0.0.1
title: AgeRange
description: Age range
type: object

meta:
  contributors:
    - description: "Jules Jacobsen"
      id: "orcid:0000-0002-3265-15918"
    - description: "Peter Robinson"
      id: "orcid:0000-0002-0736-91998"
    - description: "Michael Baudis"
      id: "orcid:0000-0002-9903-4248"
    - description: "Isuru Liyanage"
      id: "orcid:0000-0002-4839-5158"
  provenance:
    - description: Phenopackets
      id: 'https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst'
  used_by:
    - description: Phenopackets
      id: 'https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst'
  sb_status: implemented

properties:
  start:
    allof:
      "$ref": https://schemablocks.org/schemas/ga4gh/v0.0.1/Age.json
      description: Age as ISO8601 string or OntologyClass
      examples:
        - age: 'P12Y'
  end:
    allof:
      "$ref": https://schemablocks.org/schemas/ga4gh/v0.0.1/Age.json
      description: Age as ISO8601 string or OntologyClass
      examples:
        - ageClass:
            id: 'HsapDv:0000086'
            label: 'adolescent stage'
        - age: 'P16Y6M'
required:
  anyof:
    - start
    - end
examples:
  - start:
      age: 'P12Y'
      ageClass:
        id: 'HsapDv:0000086'
        label: 'adolescent stage'
    end:
      age: 'P18Y'
```

BeaconAlleleRequest beacon ↗

{S}[B] Status [i]	implemented
Provenance	○ Beacon API
Used by	○ Beacon ○ Progenetix database schema (Beacon+ backend)
Contributors	○ Marc Fiume ○ Michael Baudis ○ Sabela de la Torre Pernas ○ Jordi Rambla ○ Beacon developers...
Source (v1.1.0)	○ raw source [JSON] ○ Github

Attributes

Type: object
Description: Allele request as interpreted by the beacon.

Properties

Property	Type
alternateBases	string
assemblyId	string
datasetIds	array of string
end	integer
endMax	integer
endMin	integer
mateName	https://schemablocks.org/schemas/beacon/v1.1.0/Chror [HTML]
referenceBases	string
referenceName	https://schemablocks.org/schemas/beacon/v1.1.0/Chror [HTML]
start	integer (int64)
startMax	integer
startMin	integer
variantType	string

alternateBases

- type: string

The bases that appear instead of the reference bases. Accepted values: [ACGTN]*. N is a wildcard, that denotes the position of any base, and can be used as a standalone base of any type or within a partially known sequence. For example a sequence where the first and last bases are known, but the middle portion can exhibit countless variations of [ACGT], or the bases are unknown: ANNT the Ns can take take any form of [ACGT], which makes both ACCT and ATGT (or any other combination) viable sequences.

Symbolic ALT alleles (DEL, INS, DUP, INV, CNV, DUP:TANDEM, DEL:ME, INS:ME) will be represented in `variantType`.

Optional: either `alternateBases` or `variantType` is required.

`alternateBases` **Value Example**

assemblyId

- type: string

Assembly identifier (GRC notation, e.g. GRCh37).

`assemblyId` **Value Example**

Curie sb-vr-spec ↗

{S}[B] Status [i]	implemented
Provenance	○ vr-spec
Used by	○ vr-spec
Contributors	○ Reece Hart ○ Michael Baudis
Source (v1.0)	○ raw source [JSON] ○ Github

Attributes

Type: string
Pattern: ^\w[^:]+:.+$
Description: A string that refers to an object uniquely. T sender.

VR does not impose any contraints on strings used as ids data, the VR Specification RECOMMENDS that implement

String CURIEs are represented as `prefix:reference` (W `namespace:accession` or `namespace:local id` colloqu

The VR specification also RECOMMENDS that `prefix` be

The `reference` component is an unconstrained string.

A CURIE is a URI. URIs may *locate* objects (i.e., specify wh VR uses CURIEs primarily as a naming mechanism.

Implementations MAY provide CURIE resolution mechan

Using internal ids in public messages is strongly discoura

`Curie` Value Examples

```
"ga4gh:GA.01234abcde"
```

```
"DUO:0000004"
```

```
"orcid:0000-0003-3463-0775"
```

```
"PMID:15254584"
```

Biosample sb-phenopackets ↗

{S}[B] Status [i]	implemented
Provenance	○ Phenopackets
Used by	○ Phenopackets
Contributors	○ GA4GH Data Working Group ○ Jules Jacobsen ○ Peter Robinson ○ Michael Baudis ○ Melanie Courtot ○ Isuru Liyanage
Source (v1.0.0)	○ raw source [JSON] ○ Github

Attributes

Type: object
Description: A Biosample refers to a unit of biological material from which the substrate molec genomic DNA, RNA, proteins) for molecular analyses (e.g. sequencing, array hybridisation, mas spectrometry) are extracted.

Examples would be a tissue biopsy, a single cell from a culture for single cell genome sequencii fraction from a gradient centrifugation.

Several instances (e.g. technical replicates) or types of experiments (e.g. genomic array as well experiments) may refer to the same Biosample.

FHIR mapping: Specimen.

Properties

Property	Type
ageOfIndividualAtCollection	https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/Age.json [SRC] [HTML]
ageRangeOfIndividualAtCollection	https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/AgeRange.json [SRC] [HTML]
description	string
diagnosticMarkers	array of https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/OntologyClass.json [SRC] [HTMI
histologicalDiagnosis	https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/OntologyClass.json [SRC] [HTMI
htsFiles	array of https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/HtsFile.json [SRC] [HTML]
id	string
individualId	string
isControlSample	boolean
phenotypicFeature	array of https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/PhenotypicFeature.json [SRC] [HTML]
procedure	https://schemablocks.org/schemas/sb-phenopackets/v1.0.0/Procedure.json [SRC] [HTML]
sampledTissue	https://schemablocks.org/schemas/sb-

Checksum sb-checksum ↗

{S}[B] Status [i]	proposed
Provenance	○ GA4GH DRS (`develop` branch)
Used by	○ GA4GH DRS ○ GA4GH TRS
Contributors	○ Susheel Varma
Source (v0.0.1)	○ raw source [JSON] ○ Github

Attributes

Type: object
Description: Checksum

Properties

Property	Type
checksum	string
type	string

checksum

- type: string

The hexadecimal encoded (Base16) checksum for the data

`checksum` **Value Example**

```
"77af4d6b9913e693e8d0b4b294fa62ade6054e6b2f1ffb617ac955dd63fb0182"
```

type

- type: string

The digest method used to create the checksum. The value (e.g. `sha-256`) SHOULD be listed as `Hash Name String` in the GA4GH Hash Algorithm Registry. Other values MAY be used, as long as implementors are aware of the issues discussed in RFC6920.

GA4GH may provide more explicit guidance for use of non-IANA-registered algorithms in the future.

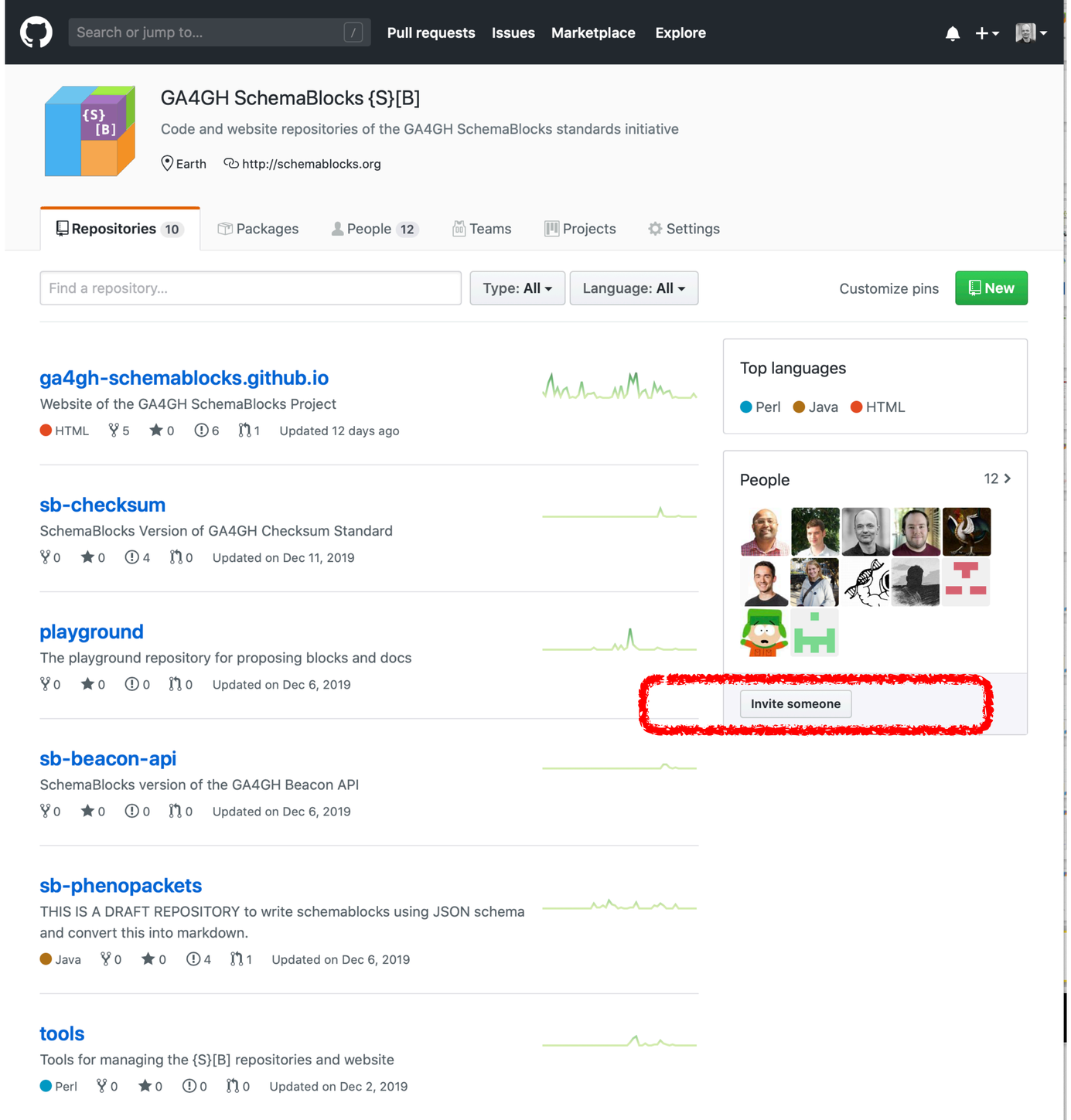
`type` **Value Example**

```
"sha-256"
```

SchemaBlocks {S}[B] - Directions & Contributions

- Recognized need of having a set of recommended standards for integrating into product development
  - ➡ no need to work through complex standards/projects like FHIR, Phenopackets ...
  - ➡ simplification of development
- SchemaBlocks {S}[B] to assume strategic position in GA4GH *TASC system
  - ➡ Inclusion into product approval processes?
  - ➡ Management/Support?
- Wish for participation of (GA4GH affiliated) groups & individuals, to ***expose*** their standards & products
- Most important role is the ***community aspect***, the interactive exchange of concepts, ideas, code, knowledge, resources ...
- Technical to-dos:
  - ➡ Lifecycle: Versioning and representation of donor schemas?
  - ➡ Development of conversion workflows for updated source products?
  - ➡ Alternative/conflicting blocks...: Graded recommendations? Name spacing?

*Technical Alignment Sub Committee



{S}[B] Info

Leads

- Melanie Courtot [↗]
- Michael Baudis [↗]

Coordination

- Melissa Konopko

Websites

- schemablocks.org
- github.com/ga4gh-schemablocks/

Meeting minutes

- schemablocks.org/categories/minutes.html