Green x Digital Consortium

Green x Digital Consortium

Technical Specifications for Data Exchange

Version 2.0

July 29, 2024

Green x Digital Consortium

Data Visualization Project

Data Format and Exchange SWG

Contents

1. Introduction

1.1. Document status

This document presents an ideal common data format and data platform for the CO2 data that should be shared across the supply chain using digital technology.

It was created by the Data Format and Exchange Sub-Working Group, which operates under the Green Digital Consortium's Visualization Working Group.

The Visualization Working Group seeks to build a mechanism to promote the visualization of CO2 data across the entire supply chain and appropriately reflect emission reduction efforts as data. As shown in Figure 1, two sub-working groups have been created under its auspices: the Methodology Sub-Working Group, which is tasked with investigating calculation methods for the CO2 data that should be shared across the supply chain using digital technology, and the Data Format and Exchange Sub-Working Group, which is tasked with examining a common data format and data platform where that CO2 data is exchanged using digital technology; and the Logistics Sub-Working Group, which is tasked with examining CO2 data calculation methods in relation to transportation and logistics.

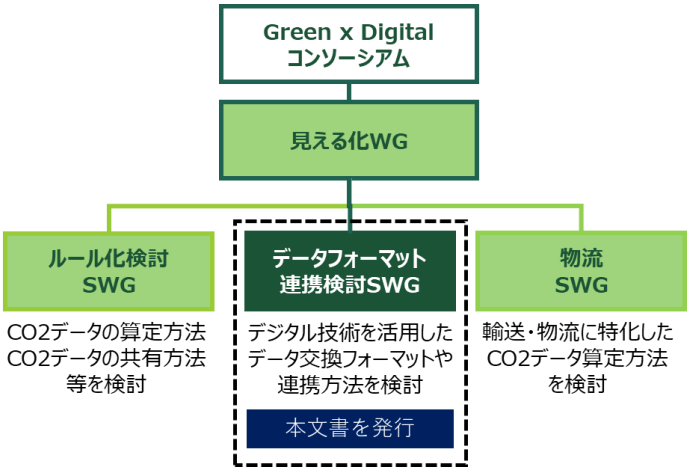

Fig. 1 Data Format and Exchange SWG and the status of this document

The main authors of this document were the leader and subleaders of the Data Format and Exchange SWG based on the results of the SWG's considerations. With cooperation from SWG members, it addresses the following:

(a) Approach to a common data format and detailed specifications
(b) Approach to a data exchange format

Members of the Data Format and Exchange SWG are shown in the following table.

Table 1 Companies comprising the Data Format and Exchange SWG

Leader	Fujitsu
Subleaders	NTT DATA, Nomura Research Institute
SWG members	IHI, Asuene, Amazon Web Services Japan, SBI R3 Japan, NTT DATA, Sustech, Zero Board, chaintope, Deloitte Tohmatsu Consulting, Toshiba, Nagase Sangyo, Nitto Denko Corporation, NEC, Nomura Research Institute, Panasonic Holdings, Hitachi, PwC Consulting, Mitsui, Ricoh

1.2. Background

There is a strong call for to achieve decarbonization throughout supply chains in all industries so as to realize carbon neutrality by 2050. This will require the accurate assessment and reduction of emissions not just by individual companies (Scope 1 and 2) but across the supply chain, including both upstream and downstream emissions (Scope 3).

Particularly for Scope 3 Category 1 (Purchased goods and services) CO2 emissions, the most common calculation method is to multiply procurement value and volume by the particular industry's average carbon intensity, but the inability of this method to reflect supplier efforts to reduce CO2 emissions has led to a growing call to acquire CO2 emissions data from suppliers. This, however, presents two major challenges.

First, the lack of a common methodology for calculating CO2 emissions that assumes the provision of data from suppliers results in disparities in data quality.

Second, while multiple carbon visualization solutions are being developed, there is no common data format or interface, etc., for exchanging data among different solutions, so when different solutions are used within the supply chain, it becomes difficult to gauge emissions across the whole chain.

Consequently, as noted above, the Green x Digital Consortium's Visualization Working Group established the Methodology SWG and the Data Format and Exchange SWG. The Methodology SWG created the "CO2 Visualization Framework," which lays out calculation methods for CO2 data to be shared using digital technology, as well as items to be disclosed when that data is shared. The Data Format and Exchange SWG considered a common data format and data platform for use when CO2 data is exchanged using digital technology, creating Version 1.0 of this document in August 2023.

Drawing on CO2 Visualization Framework Edition 1.0 and Tech Specs toward Data Exchange Version 1.0, the Green x Digital Consortium conducted a proof-of-concept experiment from late 2022 into early 2023 to confirm whether emissions data calculated by companies from a wide range of industries using a common method can be exchanged across different solutions so as to gauge supply chain carbon emissions accurately

and efficiently.1 The CO2 Visualization Framework was also revised up to Edition 2.0 accordingly. (The PoC experiment comprised two phases, with technical verification in relation to data exchange conducted in Phase 1 and practical verification conducted with the participation of user companies in Phase 2, CO2 calculation included.)

[1] PoC experiment press releases: https://www.gxdc.jp/pdf/press_release221209.pdf (Dec. 9, 2022)
https://www.gxdc.jp/pdf/press_release230215.pdf (Feb. 15, 2023)

1.3. Purpose

This document comprises Version 2.0 of the Tech Specs toward Data Exchange. Where changes are made to the specs in the frameworks referred to in this document, the document too will be updated as necessary.

1.4. Target

This document targets the CO2 data which the Green x Digital Consortium has concluded should be shared throughout the entire supply chain using digital technology.

The Methodology SWG concluded in its CO2 Visualization Framework that Green x Digital Consortium calculation methods would be set out for both product-based and organization-based calculation, taking as their premise of the product-based calculations recommended in the Pathfinder Framework developed by the WBCSD Partnership for Carbon Transparency (PACT)[2], which, like the Scope 3 Standard, are international platforms/frameworks for supply chain CO2 data exchange that have attracted attention in recent years, while also accepting the organization-based calculation that are currently in widespread use.

While this document consequently focuses on the recommended calculation data based on product data, it also adds supported attributes for organization data-based calculations, an approach unique to the Green x Digital Consortium.

As the Methodology SWG is charged with considering calculation methods for the CO2 data to be shared throughout the entire supply chain using digital technology, as well as disclosure items when data is shared, these aspects will not be addressed in this document. Please refer to the CO2 Visualization Framework issued by the Methodology SWG.

[2] WBCSD Partnership for Carbon Transparency (PACT)
The World Business Council for Sustainable Development (WBCSD) is a CEO-led community of around 200 sustainable businesses that convenes the Greenhouse Gas Protocol. The Partnership for Carbon Transparency (PACT) works under the auspices of the WBCSD to make value chain emissions more transparent and accelerate decarbonization. It defines the methodological and technical foundations necessary for emissions data exchange and issues these as part of the Pathfinder Framework and the Pathfinder Network. The Green x Digital Consortium participates in the PACT ecosystem as a collaborator. Website: https://www.carbon-transparency.com/

2. Common data format

2.1. Purpose

Accurately gauging CO2 emissions throughout the entire supply chain (particularly Scope 3 emissions) will be vital in achieving supply chain decarbonization on a global basis. In particular, while using primary data from supplier firms is vital in calculating Scope 3 emissions so as to reflect supplier efforts to reduce emissions, the data provided by these suppliers is currently quite diverse according to recipient company use cases and to the CO2 visualization solutions used by the various companies supplying and receiving that data. At the same time, the WBCSD PACT Pathfinder Framework and Pathfinder Network are leading the way internationally in terms of frameworks for supply chain CO2 data exchange, and consideration must be given to interoperability internationally as well as among different types of systems so as to reflect activity across the global span of supply chains.
Consequently, in this document, the Data Format and Exchange SWG set out to provide a common data format contributing to the communication of CO2 data that is within the means of small and medium enterprises in the supply chain and that will enable international data exchange and linkage.

2.2. Policy

One method of creating a common data format would be to analyze data items in existing visualization solutions and set these items as the least common multiples. Another would be to create a common data format to realize the content of the CO2 Visualization Framework produced from Methodology SWG considerations. In addition, the Visualization Working Group is examining data utilization, and yet another approach would be to prepare common data items to realize that data utilization. Looking internationally, WBCSD PACT released the Pathfinder Network Technical Specifications for PCF Data Exchange Version 2.0.13 [3] on September 27, 2023 and Version 2.1.0 on December 7, 2023, laying out common data items for data sharing. Given the above, these guidelines select common data items based on the following perspectives:

- Data items to be based on the content of the CO2 Visualization Framework produced by the Methodology SWG.

[3] Version 2.0.0 was released on February 21, 2023 following the announcement of Pathfinder Framework Version 2 on January 25 that year. Version 2.0.1 of the Tech Specs was subsequently released on September 27, 2023 and Version 2.1.0 on December 7, 2023.
Version 2.0.0: https://wbcsd.github.io/tr/2023/data-exchange-protocol-20230221/
Version 2.0.1: https://wbcsd.github.io/tr/2023/data-exchange-protocol-20230927/
Version 2.1.0: https://wbcsd.github.io/tr/2023/data-exchange-protocol-20231207/

- Because supply chains are global in nature, items to be consistent with international activities. Specifically, consider linkage with the PACT Technical Specifications for PCF Data Exchange and related documents.
- Consideration also to be given to data utilization by solution users.

2.3. Common data format details

2.3.1. Global items and unique items

The common data format is divided into global items and items unique to the Green x Digital Consortium.

The global items generally follow the data model defined in Pathfinder Network Technical Specifications for PCF Data Exchange Version 2.1.0 (issued December 7, 2023; below, the Pathfinder Network data model), but also include the Green x Digital Consortium's own specifications in some places (see subsections below for further information).

Items unique to the Green x Digital Consortium comprise those which we have added with a view to data utilization in the CO2 Visualization Framework, the utility of which was tested during Phase 2 of our PoC experiment based on Pathfinder Network Technical Specifications Version 1.0.0 (issued June 16, 2022) (see the PoC experiment press releases noted above).

Please note the following in relation to item details:

- The three requirement levels are "Mandatory," "Optional," and "Recommendation." Mandatory items must be included in the data.
- If a company is unable to meet the mandatory disclosure requirements for certain items, it MAY still exchange data as long as this non-compliance is disclosed. Another possible approach would be to exchange in non-compliant data exchange, to the extent that this is only the provision of data from upstream suppliers.
- Optional items marked with an asterisk are currently optional but will become mandatory as of 2025 (DQIs, etc.)
- Because the JSON format is used for data in HTTP payloads, the respective JSON data types are noted. (In the case of object-type data, include a link to the details in the footnotes.)
- For items that allow decimals (fossilGhgEmissions, etc.), we have provisionally specified up to 16 significant digits.
- Refinements made as a result of our inquiries to PACT as well as issues for future consideration have been flagged in notes.
- Those Green x Digital Consortium specifications included in global items are underlined.

2.3.2. Global items

The particulars of the Pathfinder Network data model are as follows. For specification details, please refer to "Pathfinder Network Technical Specifications Version 2.1.0.[4]" (issued December 7, 2023).

2.3.2.1. id

Requirement level: Mandatory

JSON data type: String

Product Footprint identifier. Used to identify individual data linked across platforms. MUST be a UUID (Universally Unique Identifier) v4 as specified in RFC 4122.

2.3.2.2. specVersion

Requirement level: Mandatory

[4] PCF Data Exchange version 2.1.0
https://wbcsd.github.io/tr/2023/data-exchange-protocol-20231207/

JSON data type: String

Version of the data model specifications managed by PACT. In Version 2.1.0 a string expressed with value “2.1.0”.

2.3.2.3. precedingPfIds

Requirement level: Optional
JSON data type: Array of strings

An array in which the elements are product footprint identifiers deprecated as a result of major updates, etc. If defined, MUST be a non-empty set without duplicates.

2.3.2.4. version

Requirement level: Mandatory
JSON data type: Number

Product Footprint version. The point immediately after data generation is 0, increasing in increments each time there is an update.

2.3.2.5. created

Requirement level：Mandatory
JSON data type: String

Datetime of data generation on the platform. Datetime string must match ISO 8601 format. (Time zone must be UTC.)

2.3.2.6. updated

Requirement level: Optional
JSON data type: String

Datetime of data updating on the platform. Datetime string must match ISO 8601 format. (Time zone must be UTC.)

A ProductFootprint SHOULD include the property updated with value the timestamp of the ProductFootprint update. A ProductFootprint MUST NOT include this property if an update has never

been performed.

2.3.2.7. status

Requirement level: Mandatory
JSON data type: String

String indicating whether or not each ProductFootprint can be used. MUST include the property status with value one of the following values:

Active
The default status of a product footprint is Active. A product footprint with status Active can be used by data recipients for product footprint calculations.
Deprecated
The product footprint is deprecated and SHOULD NOT be used for product footprint calculations by data recipients.

2.3.2.8. statusComment

Requirement level: Optional
JSON data type: String

Comment explaining the reason for the current status.

2.3.2.9. validityPeriodStart

Requirement level: Optional
JSON data type: String

The start date of the time interval during which the ProductFootprint is declared as valid for use by a data recipient (a string compliant with ISO8601). If a validity period is not specified, the ProductFootprint shall be valid for 3 years. If the validity period is to be specified, then the value of validityPeriodStart MUST be defined with value greater than or equal to the value of referencePeriodEnd. The value of validityPeriodEnd MUST be defined with value strictly greater than validityPeriodStart, and less than or equal to referencePeriodEnd + 3 years.

* Vallidity period is inclusive of the date of validityPeriodStart but exclusive of the date of validityPeriodEnd (noted below).

2.3.2.10. validityPeriodEnd

Requirement level: Optional
JSON data type: String

The end date of the time interval during which the ProductFootprint is declared as valid for use by a data recipient (a string compliant with ISO8601).

2.3.2.11. companyName

Requirement level: Mandatory

JSON data type: String

String indicating the name of the data owner. Used with the aim of supplementing owner identification through companyIds (see below). String of length 1 or greater.

2.3.2.12. companyIds

Requirement level: Mandatory
JSON data type: Array<String>

A non-empty set of URN (Uniform Resource Names), each value of which uniquely identifies the data owner.
[Note]
Company ID that uniquely identifies that company, such as its DUNS Number, ISIN, or ticker code. For the URN, companies can use any namespace registered with IANA such as the OID (Object Identifier) or GLN (Global Location Number) that uniquely identifies the company globally.
Extended specifications are currently being created at PACT to deal with industry-specific identifiers, etc., that are not IANA-registered.

2.3.2.13. productDescription

Requirement level: Mandatory
JSON data type: String

Free-form description of the product plus other information related to it such as production technology or packaging. Can be left empty.

2.3.2.14. productIds

Requirement level: Mandatory
JSON data type: Array<String>

A non-empty array containing as an element the URN (Uniform Resource Name) which uniquely identifies the data user.
［Note］
For the URN, companies can use any namespace registered with IANA, such as the OID (Object Identifier) or GTIN (Global Trade Item Number) that uniquely identifies the product globally.
Extended specifications are currently being created at PACT to deal with industry-specific identifiers and in-house coding systems, etc., that are not IANA-registered.

productCategoryCpc

Requirement level: Mandatory
JSON data type: String

UN CPC Code (Central Product Classification Code) version 2.1 that the given product belongs to. Can be left empty.

2.3.2.15. productNameCompany

Requirement level: Mandatory
JSON data type: String

Trade name of the product. Used to supplement product identification through the productIds noted above (e.g. information on production customization). String of length 1 or greater.

2.3.2.16. comment

Requirement level: Mandatory
JSON data type: String

Used for information and instructions related to the calculation of the footprint, or other information which informs the ability to interpret, to audit or to verify the product footprint. Can be left empty.

[Note]
- A future issue for consideration will be whether, if audit and verification methods are to be identified, properties should be defined for ease of interpretation as a system rather than comments being left free-form.

2.3.2.17. extensions

Requirement level: Optional
JSON data type: Array of strings

See the following link for format details:
https://wbcsd.github.io/data-model-extensions/spec/

2.3.2.18. pcf

Requirement level: Mandatory
JSON data type: Object

The product carbon footprint identified according to the above-noted properties and related data. Includes the various properties noted below.

2.3.2.18.1. declaredUnit

Requirement level: Mandatory

JSON data type: String

Any of the following values can be set.

- liter

for special SI Unit litre

- kilogram

for SI Base Unit kilogram

- cubic meter

for cubic meter, the Derived Unit from SI Base Unit metre

- kilowatt hour

for kilowatt hour, the Derived Unit from special SI Unit watt

- megajoule

for megajoule, the Derived Unit from special SI Unit joule

- ton kilometer

for ton kilometer, the Derived Unit from SI Base Units kilogram and metre

- square meter

for square meter, the Derived Unit from SI Base Unit metre

Green x Digital Consortium specification:

- yen: ¥

For the unit definition, see Bureau International des Poids et Mesures.[5]

［Note］

- When we asked PACT about the inclusion of the product unit as a value, PACT noted that because this might introduce subjectivity, PACT had excluded it with a view to maintaining comparability. Because some Green x Digital Consortium member companies want product unit included, however, we will continue to consult with PACT on this issue.

2.3.2.18.2. unitaryProductAmount

Requirement level: Mandatory
JSON data type: String

The declaredUnit amount. MUST be expressed as a decimal number equal to or greater than 0.

2.3.2.18.3. pCfExcludingBiogenic

Requirement level: Mandatory
JSON data type: String

The product carbon footprint of the product excluding biogenic CO2 emissions. The value MUST be calculated per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal equal to or greater than zero.

[5] Bureau International des Poids et Mesures (Table 8. Non-SI units accepted for use with the SI units)
https://www.bipm.org/documents/20126/41483022/SI-Brochure-9-EN.pdf/2d2b50bf-f2b4-9661-f402-5f9d66e4b507

2.3.2.18.4. pCfIncludingBiogenic

equirement level: Optional*
JSON data type: String

The product carbon footprint of the product including all biogenic emissions (CO2 and otherwise). The value MUST be calculated per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal. Can be less than zero.

2.3.2.18.5. fossilGhgEmissions

Requirement level: Mandatory
JSON data type: String

The emissions from fossil sources as a result of fuel combustion, from fugitive emissions, and from process emissions (greenhouse gases emitted during the chemical transformation of materials). MUST be calculated per declared unit with unit kg of CO2 equivalent per kilogram (kgCO2e/kg), expressed as a decimal number equal to or greater than zero. See the Pathfinder Framework or the CO2 Visualization Framework for the calculation methodology.

2.3.2.18.6. fossilCarbonContent

Requirement level: Mandatory
JSON data type: String

The fossil carbon content of the product (kgC / declaredUnit). The value MUST be calculated per declared unit with unit kg Carbon per declared unit (kgC / declaredUnit), expressed as a decimal equal to or greater than zero.

2.3.2.18.7. biogenicCarbonContent

Requirement level: Mandatory
JSON data type: String

The biogenic carbon content of the product (kgC / declaredUnit). The value MUST be calculated per declared unit with unit kg Carbon per declared unit (kgC / declaredUnit), expressed as a decimal equal to or greater than zero.

2.3.2.18.8. dLucGhgEmissions

Requirement level: Optional*
JSON data type: String

Emissions resulting from carbon stock loss due to land conversion on the area of land under consideration. The value of this property MUST include direct land use change (dLUC) where available, otherwise statistical land use change (sLUC) can be used. The value MUST be calculated

per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal equal to or greater than zero.

2.3.2.18.9. landManagementGhgEmissions

Requirement level: Optional*
JSON data type: String

GHG emissions and removals associated with land-management-related changes, including non-CO2 sources. The value MUST be calculated per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal. Can be less than zero.

2.3.2.18.10. otherBiogenicGhgEmissions

Requirement level: Optional*
JSON data type: String

All other biogenic GHG emissions associated with product manufacturing and transport that are not included in dLUC, iLUC, and land management (landManagementGhgEmissions). The value MUST be calculated per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal equal to or greater than zero.

2.3.2.18.11. iLucGhgEmissions

Requirement level: Optional
JSON data type: String

Emissions resulting from recent (i.e., previous 20 years) carbon stock loss due to land conversion on land not owned or controlled by the company or in its supply chain. The value MUST be calculated per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal equal to or greater than zero.

2.3.2.18.12. biogenicCarbonWithdrawal

Requirement level: Optional*
JSON data type: String

The biogenic carbon contained in the product converted to kilogram of CO2e. The value MUST be calculated per declared unit with unit kgCO2e / declaredUnit expressed as a decimal equal to or less than zero.

2.3.2.18.13. aircraftGhgEmissions

Requirement level: Optional
JSON data type: String

The GHG emissions resulting from aircraft engine usage for the transport of the product. The value

MUST be calculated per declared unit with unit kg of CO2 equivalent per declared unit (kgCO2e / declaredUnit), expressed as a decimal equal to or greater than zero.

2.3.2.18.14. characterizationFactors

Requirement level: Mandatory
JSON data type: String

The IPCC version of the GWP characterization factors used in PCF calculation. The value MUST be either AR6 (Sixth Assessment Report of the IPCC) or AR5 (Fifth Assessment Report of the IPCC).

crossSectoralStandardsUsed

Requirement level: Mandatory
JSON data type: Array of strings

The cross-sectoral standards applied for calculating or allocating GHG emissions. Some or all of the following strings MUST be entered (cannot be an empty array).

- GHG Protocol Product standard
- ISO Standard 14067
- ISO Standard 14044

Green x Digital Consortium specification:

- ISO Standard 14040
- ISO Standard 14025
- ISO T/S 14027
- Green x Digital Product Framework (CO2 Visualization Framework Product-based calculation)
- GHG Protocol Scope 3 Standard
- Green x Digital Scope 3 Framework (CO2 Visualization Framework Organization-based calculation)

* While this item assumes standards such as the above, because this is a free-form entry in which other values are also inserted, according to the value of the item, systemic checks are not undertaken of organization-based and product-based calculations, etc.

2.3.2.18.15. productOrSectorSpecificRules

Requirement level: Optional
JSON data type: Array of objects

Rules used for calculating or allocating GHG emissions (PCRs, etc.). An array of objects with the following properties as its elements. If no product or sector specific rules were followed, this set MUST be empty.

Green x Digital Consortium specification:
Leave empty in the case of organization-based calculations.

［Note］

- We will consider whether what to enter for otherOperatorName and ruleNames should be

defined when Japanese PCRs are applied.

2.3.2.18.15.1. operator

Requirement level: Mandatory
JSON data type: String

One of the following strings MUST be enumerated:

- PEF
  - For EU PEF Methodology PCRs
  - For EU Product Environmental Footprint
- EPD International
  - For PCRs authored or published by EPD International
- Other
  - For PCRs *not* published by the above operators

2.3.2.18.15.2. ruleNames

Requirement level: Mandatory
JSON data type: Array of strings

Non-empty array with elements comprising a non-empty string.

1.1.1.1.1.1. otherOperatorName

Requirement level: Optional
JSON data type: NonEmptyString

If the value of property operator is Other, a ProductOrSectorSpecificRule MUST include the property otherOperatorName. In this case, the operator declared MUST NOT be included in the definition of ProductOrSectorSpecificRuleOperator. If the value of property operator is not Other, the property MUST be undefined.

2.3.2.18.16. biogenicAccountingMethodology

Requirement level: Optional*
JSON data type: String

The standard followed to account for biogenic emissions and removals. The value MUST be one of the following:

- PEF: For the EU Product Environmental Footprint Guide
- ISO: For the ISO 14067 standard
- GHGP: For the Greenhouse Gas Protocol (GHGP) Land Sector and Removals Guidance
- Quantis: For the Quantis Accounting for Natural Climate Solutions Guidance

2.3.2.18.17. boundaryProcessesDescription

Requirement level: Mandatory
JSON data type: String

The processes attributable to each lifecycle stage. Can be left empty.

Example: Electricity consumption included as an input in the production phase

2.3.2.18.18. referencePeriodStart

Requirement level: Mandatory
JSON data type: String

The reporting start date. Must be a string compliant with ISO8601 (UTC time zone). The maximum validity period shall be up to three years, provided that no major changes to the production process take place within that period. Major changes are defined as a variance of 10 percent or more compared to the original PCF. See the CO2 Visualization Framework or Pathfinder Framework section 6.1.2.1 for further details.[6]

2.3.2.18.19. referencePeriodEnd

Requirement level: Mandatory
JSON data type: String

The reporting end date. Must be a string compliant with ISO8601 (UTC time zone). See the CO2 Visualization Framework or the Pathfinder Framework for further details (same as for referencePeriodStart).

2.3.2.18.20. geographyCountrySubdivision

Requirement level: Optional
JSON data type: String

This property MUST be defined if the CarbonFootprint has a country subdivision-specific geographical granularity, (e.g., Tokyo). (In such cases, the properties geographyCountry and geographyRegionOrSubregion must be undefined.) An ISO 3166-2 subdivision code can be used.

[Note]
In the case of Tokyo, the local government code is JP-13.

2.3.2.18.21. geographyCountry

Requirement level: Optional
JSON data type: String

This property MUST be defined if the CarbonFootprint has a country-specific geographical granularity (e.g., Japan). In such cases, the properties geographyCountrySubdivision and geographyRegionOrSubregion must be undefined. An ISO 3166-1 alpha-2 country code can be used (e.g., JP).

2.3.2.18.22. geographyRegionOrSubregion

[6] https://www.carbon-transparency.com/media/srhhloun/pathfinder-framework.pdf

Requirement level: Optional
JSON data type: String

This property MUST be defined if the CarbonFootprint has a regional or sub-regional geographical granularity (e.g., Asia). In such cases, the properties geographyCountrySubdivision and geographyCountry must be undefined. See the example of Asia in Pathfinder Network Technical Specifications v2.0.1 at:
https://wbcsd.github.io/data-exchange-protocol/v2/#enumdef-regionorsubregion

1.1.1.1.1. secondaryEmissionFactorSources

Requirement level: Optional
JSON data type: Array of objects

If set, MUST be a non-empty array in which elements comprise objects including the following properties. If not set, this property MUST be undefined.

2.3.2.19.24.1 name

Requirement level: Mandatory (where there are emissionFactorSources)
JSON data type: String

The non-empty name of the emission factor database.

2.3.2.19.24.2. version

Requirement level: Mandatory (where there are emissionFactorSources)
JSON data type: String

The non-empty version of the emission factor database.

1.1.1.1.2. exemptedEmissionsPercent

Requirement level: Mandatory
JSON data type: Number

Percentage of emissions excluded from PCF. Cutoff rate (exclusion of emissions that do not need to be considered). Expressed as a decimal number between 0.0 and 5.

1.1.1.1.3. exemptedEmissionsDescription

Requirement level: Mandatory
JSON data type: String

Rationale behind exclusion of specific PCF emissions, can be an empty string if no emissions were excluded.

1.1.1.1.4. packagingEmissionsIncluded

Requirement level: Mandatory
JSON data type: Boolean

A boolean flag indicating whether packaging emissions are included in the PCF (pCfExcludingBiogenic, pCfIncludingBiogenic) (true or false).

1.1.1.1.5. packagingGhgEmissions

Requirement level: Optional
JSON data type: String

Emissions resulting from the packaging of the product. The value MUST be calculated per declared unit with unit kg of CO2 equivalent per kilogram (kgCO2e / declared unit), expressed as a decimal equal to or greater than zero.

1.1.1.1.6. allocationRulesDescription

Requirement level: Optional
JSON data type: String

Description of any allocation rules applied.

2.3.2.19.30. uncertaintyAssessmentDescription

Requirement level: Optional
JSON data type: String

The results, key drivers, and a short qualitative description of the uncertainty assessment. Where the primary data is not certain (unreliable, unstable values, etc.), there are cases where secondary data may be used, with this noted in the comments.

2.3.2.19.31. primaryDataShare

Requirement level: Optional*
JSON data type: Number

The share of primary data as a percentage, expressed as a decimal number between 0 and 100.

Green x Digital Consortium specification:
In the case of organization-based calculations, enter “0”.

For reporting periods ending before the beginning of year 2025, at least property primaryDataShare or propery dqi MUST be defined.

For reporting periods including the beginning of year 2025 or after, this property MUST be defined.

2.3.2.19.32. dqi

Requirement level: Optional*
JSON data type: Object

Data Quality Indicators. For reporting periods ending before the beginning of year 2025, at least property primaryDataShare or propery dqi MUST be defined.[7]

For reporting periods including the beginning of year 2025 or after, this property MUST be defined.

2.3.2.19.33. assurance

Requirement level: Optional
JSON data type: Object

Assurance information in accordance with the Pathfinder Framework.[8]

[7] For details on dqi, see 4.2.3: Data quality assessment in the Pathfinder Framework. https://www.carbon-transparency.com/media/srhhloun/pathfinder-framework.pdf
[8] Tech Specs 4.4. Data Type: Assurance https://wbcsd.github.io/data-exchange-protocol/v2/#elementdef-assurance

1.1.2. Green x Digital Consortium items

2.3.3.1. calcMethod【unique to the Green x Digital Consortium】

Requirement level: Mandatory
JSON data type: String

Set one of the following values:
Product-based calculation: 0
Organization-based calculation: 1

2.3.3.2. FWedition 【unique to the Green x Digital Consortium】

Requirement level: Mandatory
JSON data type: String

The version of the Carbon Visualization Framework referenced. In the case of Carbon Visualization Framework Edition 2.0, enter “2.0”.

1.1.2.1. gateToGate【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: Object

Entered, gate-to-gate emissions can also be provided in addition to cradle-to-gate.

2.3.3.3.1. gateToGateFossilGhgEmissions【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

The company’s own emissions from the gate-to-gate boundary, provided as a subset of total emissions (excluding biogenic emissions and removals), expressed as kg-CO2e.

2.3.3.3.2. gateToGateBiogenicCarbonContent【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

Gate-to-gate biogenic carbon content (kg-CO2/declaredUnit).
A company’s own emissions, excluding the upstream portion, provided as a subset of the figure in 2.3.2.19.7. biogenicCarbonContent.

2.3.3.3.3. unitaryProductAmountComment 【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

Free-form entry adding an explanation of the product amount and declared unit.

[Note]
In Organization-based calculations, where the currency unit (yen, dollar, etc.) is the declared unit, because it is difficult to distinguish product-unit CO2 data and transaction-unit CO2 data, this is explained in this column.

1.1.2.2. certificateAmount 【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

Certificate amount per declared unit (kg-CO2eq/declaredUnit).
Add a report on the amount purchased and adjusted by the data disclosing party.

2.3.3.5. certificateType【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

Examples:
- Renewable energy certificates
- J Credits
- Non-fossil fuel certificates

1.1.2.3. scope3Category 【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: Array<Number>

For organization-based calculations: In addition to Scope 1 and 2, enter Scope 3 calculation categories as an array (e.g. 1, 4, 5).

2.3.3.7. scope1DistributionLevel 【unique to the Green x Digital Consortium】

Requirement level: Mandatory
JSON data type: String

2.3.3.8. scope2DistributionLevel 【unique to the Green x Digital Consortium】

Requirement level: Mandatory
JSON data type: String

For organization-based calculations: Scope 1-3 physical distribution level (one of the following):

- Organization
- Site
- Building
- Production line

2.3.3.9. scope1DistributionIndex 【unique to the Green x Digital Consortium】

Requirement level: Recommendation

JSON data type: String

2.3.3.10. scope3DistributionLevel 【unique to the Green x Digital Consortium】

Requirement level: Mandatory

JSON data type: String

For organization-based calculations: Scope 1-3 physical distribution level (one of the following):

- Organization
- Site
- Building
- Production line

* Where there are multiple Scope 3 levels, enter the predominant level.

2.3.3.11. distributionLevelComment 【unique to the Green x Digital Consortium】

Requirement level: Optional

JSON data type: String

Free-form entry for adding further details about the distribution level.

2.3.3.12. scope2DistributionIndex【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

2.3.3.13. scope3DistributionIndex【unique to the Green x Digital Consortium】

Requirement level: Recommendation
JSON data type: String

For organization-based calculations: Index for Scope 1-3 physical distribution level (one of the following):

- Volume
- Unit
- JPY

2.3.3.14. distributionIndexComment【unique to the Green x Digital Consortium】

Requirement level: Optional
JSON data type: String

Free-form entry for adding further details about the distribution index.

2. Data linkage platform

2.1. Approach

Based on SWG investigations into the ideal data linkage platform, we decided on the following approach to the platform which should be developed by the Green x Digital Consortium.

The first digital linkage platform model that we considered was the centralized database model shown below from the Green x Digital Consortium Visualization Working Group report entitled "Study Preparation Phase/Primary Report for Establishment of Mechanism for Visualization of Supply Chain CO2".[9] In this model, companies participating in the Consortium share a database and server. The merits of the model are the unified data structure and the centralized control of visibility and tamper resistance, but the maintenance-related cost burden that it would place on the Consortium would need to be considered, as well as the operating scheme.

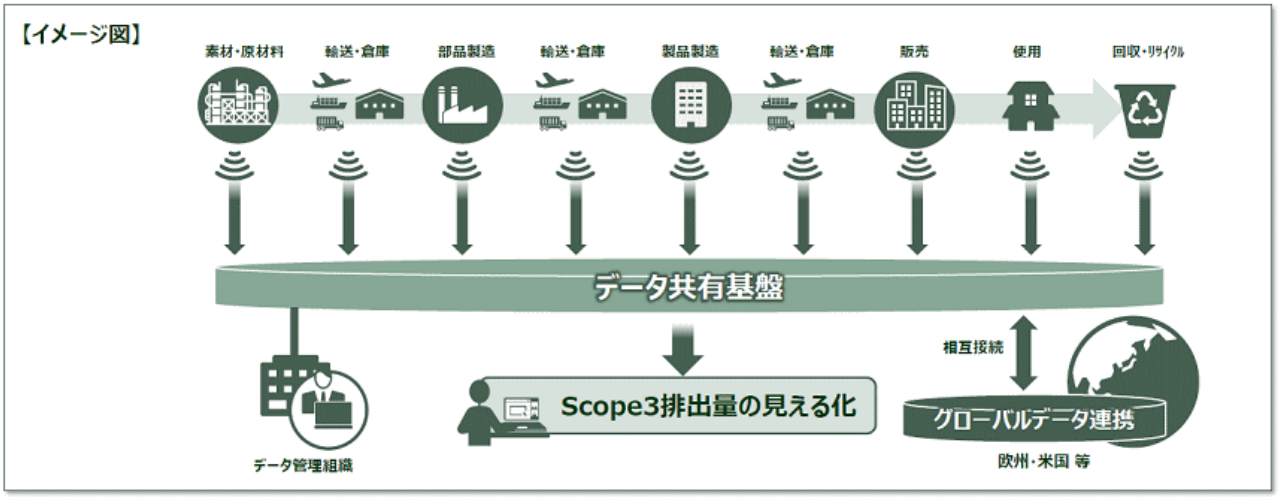

Fig. 2 Data-sharing platform model in primary report

Next, the SWG looked at adopting a non-centralized control model, examples of which are shown below. However, user companies have already adopted various different visualization solutions and have business policies tailored to specific solution vendors, while linking visualization solutions and data linkage solutions would require massive system enhancement on the part of the solution providers. Consequently, this model was not deemed to be the ideal platform for the Green x Digital Consortium to develop.

[9] https://www.gxdc.jp/pdf/achievement_report.pdf

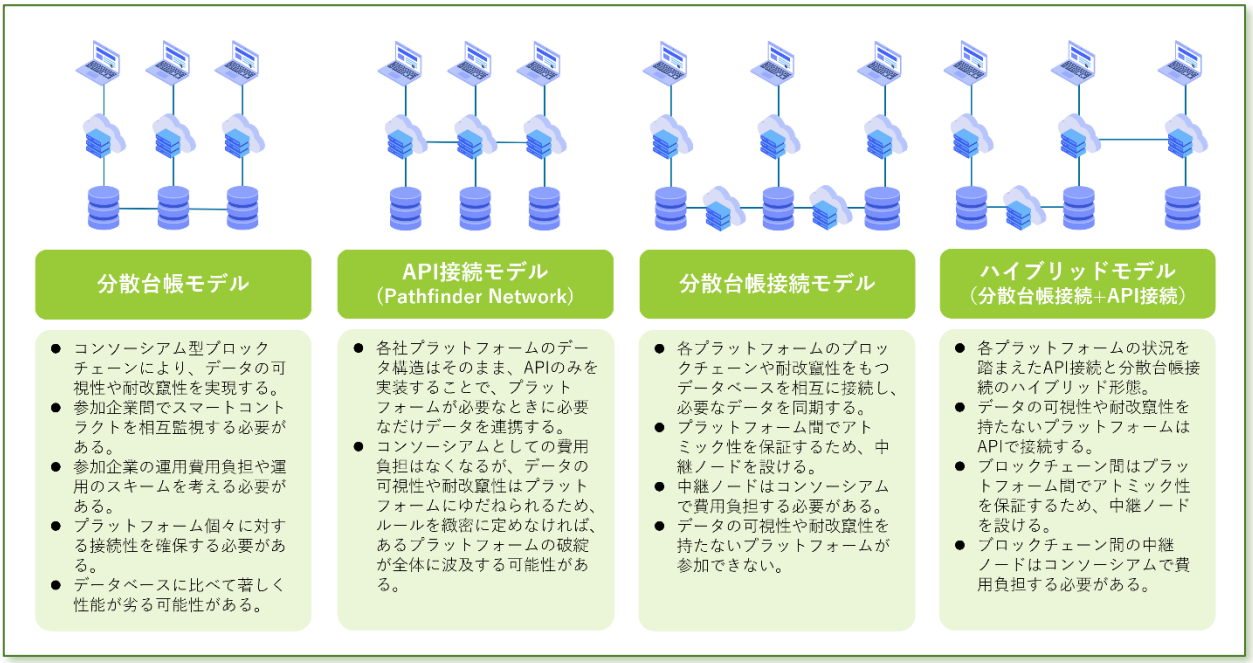

Fig. 3 Image of non-centralized control model

Both models also pose major cost issues for small and medium enterprises in the supply chain, making it difficult to achieve data linkage throughout the entire supply chain. We consequently reached the following conclusions in terms of the ideal platform for the Green x Digital Consortium:

- To develop only a common data format, not a common data platform
- To leave the decision on which data acquisition method is used among the companies making up a supply chain as a B2B decision. (The Pathfinder Network assumes API linkage, and in this document too, specifications in the case of API linkage are noted in the next section. An API connection model will also be used in the PoC experiment, but we will not position this as the only data linkage method.

2.2. Data linkage specifications

This section notes the envisaged pattern in the case of data linkage between visualization solutions (product carbon footprint calculation applications, etc.) and data linkage solutions (platform services offering traceability and tamper resistance when engaging in data linkage) as well as specifications for API linkage.

An API has already been released by PACT as part of the Pathfinder Network for exchanging and sharing product carbon footprint information between different solutions. Given global common needs in relation to GHG emissions, in the case of API linkage, as the first step, we envisage data exchange either through the Pathfinder Network API and, for the Green x Digital Consortium's additional items, the Consortium's own API.

Next step, we envisage data exchange through the Pathfinder Network API for those data elements that are relevant across industries and regions. For those data elements that are Green x Digital unique, we envisage the use of a Pathfinder Network data model extension over which these elements will be shared.

a

2.2.1. Envisaged data linkage pattern

There are three possible patterns for data linkage combinations: (1) direct peer-to-peer linkage between visualization solutions; (2) visualization solutions linking to a data linkage solution of some nature; and (3) linkage between data linkage solutions. We also assume two linkage methods, one using an API and the other not.

2.2.1.1. Direct peer-to-peer linkage between visualization solutions

As this format is consistent with PACT assumptions, solution providers can realize data linkage using common specifications by using either the Pathfinder Network API and the Green x Digital Consortium's own API. However, because information on what product is managed on what platform is not supplied by either API, product and solution mapping data needs to be managed external to the solutions (or standardized internally).

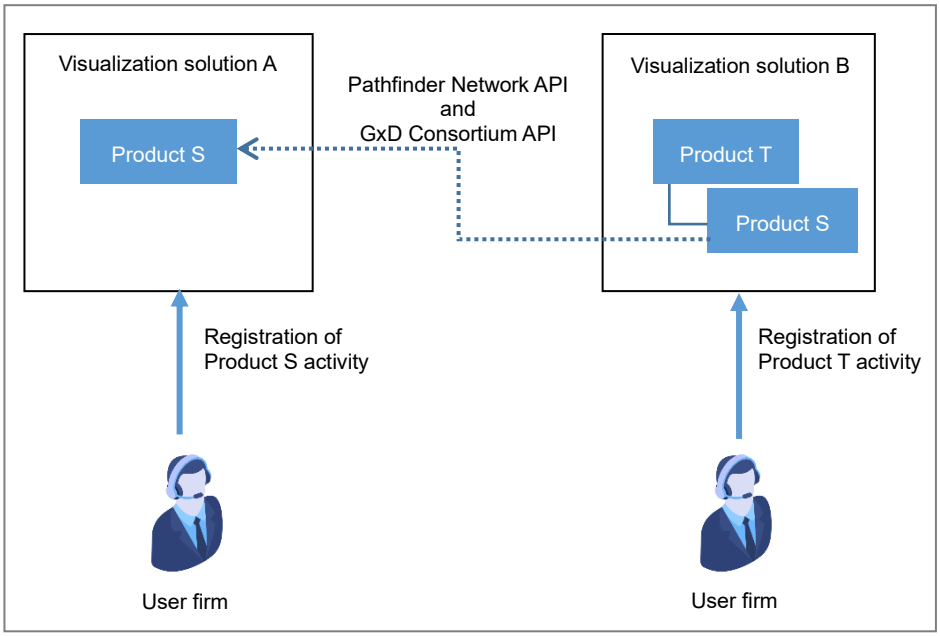

Fig. 4 Direct peer-to-peer linkage of visualization solutions

2.2.1.2. Visualization solutions linking to a data linkage solution of some nature

As noted above, data linkage between visualization solutions requires mapping data external to the solutions. By serving this role, data linkage solutions can reduce the cost of data management for visualization solution linkage (only the connection to the data linkage solution needs to be considered). When the scale of data linkage increases, networks between solutions become more complex and network management costs arise, but these costs can be alleviated through centralized connection management by the data linkage solution. At the same time, where the data linkage solution experiences telecommunications issues for some reason, the whole network can break down, which can greatly boost the asymmetric costs of data linkage solutions.

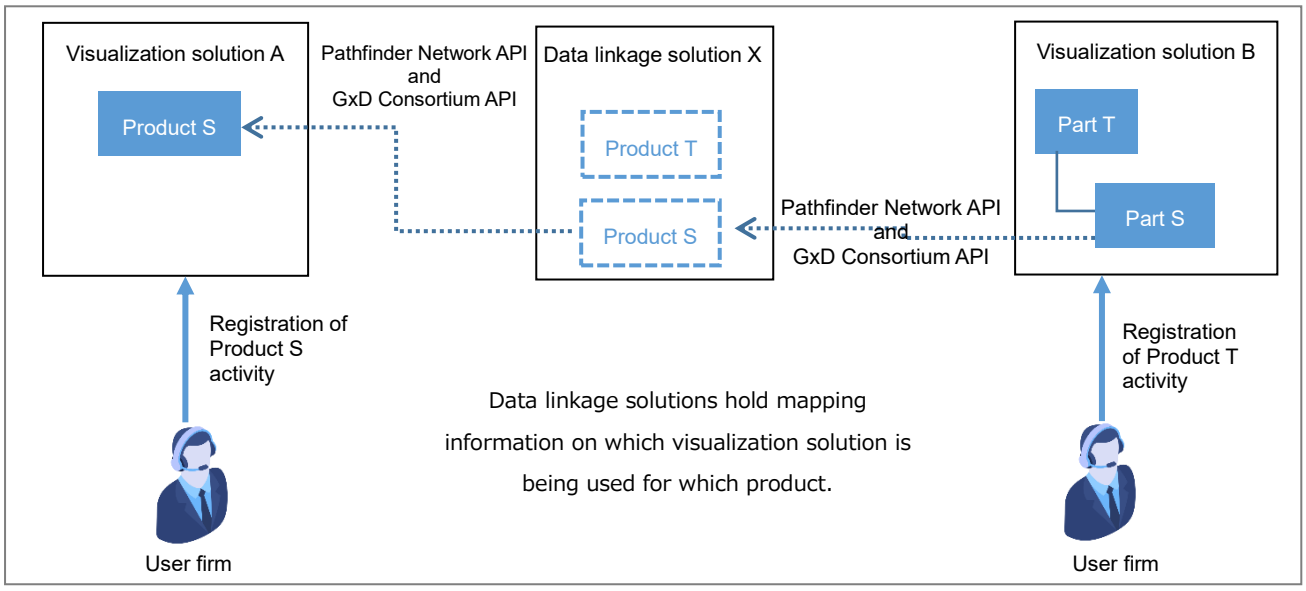

Fig. 5 Visualization solutions linking to a data linkage solution of some nature

2.2.1.3. Linkage between data linkage solutions

As noted above, the network structure can be simplified by using data linkage solutions for data linkage between visualization solutions but data linkage solution costs can also soar. To remove this problem, multiple data linkage solutions can be prepared and interoperated so as to boost tamper resistance and distribute the burden. However, to realize this, further consideration needs to be given to the standardization of methods of synchronizing information among data linkage solutions.

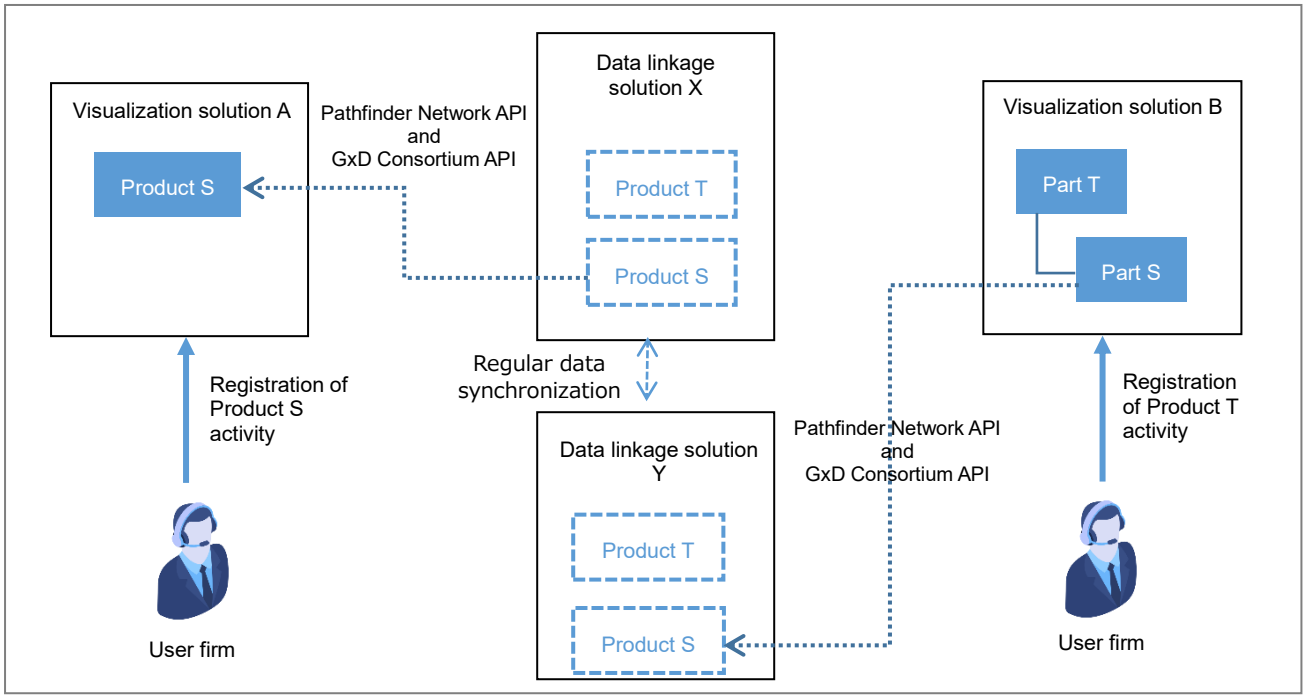

Fig. 6 Linkage between data linkage solutions

2.2.2. Pathfinder Network API

The Pathfinder Network API will be used for API linkage in relation to global items.
For Pathfinder Network API specifications, see the HTTP REST API in Chapter 6 of the Version Pathfinder Network Technical Specifications for PCF Data Exchange (Version 2.1.0), issued December 7, 2023.

2.2.3. Green x Digital Consortium API

The Green x Digital Consortium's own API will be used for API linkage in relation to items unique to the Green x Digital Consortium. With a view to future interoperability with the Pathfinder Network API, these are limited to additions to the Consortium's own model, with Pathfinder Network API technical specifications referenced to the greatest extent possible.

A host system MUST implement actions Action Authenticate, Action ListFootprints, and Action GetFootprint, and should implement Action Events.

A host system MUST offer its actions under https method only. It should offer an OpenId Provider Configuration Document as specified in [OPENID-CONNECT].

A host system MUST offer all actions under the same Hostname and Subpath except for the token endpoint (Action Authenticate and the endpoint returned from the OpenId Provider Configuration Document).

A host system can offer the OpenId Provider Configuration Document and Action Authenticate under AuthHostname and AuthSubpath which are different from Hostname and Subpath.

If a host system does not offer an OpenId Provider Configuration Document, data recipients MUST assume that Action Authenticate is offered under AuthHostname/AuthSubpath/auth/token.

The host system's DNS domain name is example.org and the subpath is /wbcsd whereas the ID management system uses a id.example.org domain with an empty subpath. The URIs would then be:

- for OpenId Provider Configuration Document: https://id.example.org/.well-known/openid-configuration
- for Action Authenticate: https://id.example.org/auth/token
  for Action ListFootprints: https://example.org/wbcsd/2/footprints, etc.

2.2.3.1. Authenticate

Request an access token using client credentials.

Host systems MUST implement this action in conformance with [rfc6749] Section 4.4 (OAuth2 Client Credentials). Host systems MAY offer this action under a dedicated AuthHostname and AuthSubpath.

2.2.3.1.1. Request syntax

For reasons of backwards-compatibility with data recipients implementing the Version 2.0 authentication flow only, Host systems MUST offer this action under path AuthSubpath/auth/token and hostname AuthHostname.

```
POST AuthSubpath/auth/token HTTP/1.1
host: AuthHostname
accept: application/json
content-type: application/x-www-form-urlencoded
authorization: Basic BasicAuth
content-length: ContentLength
```

AuthBody

In addition, if a host system supports OpenId Connect, the host system can offer and implement this Action under a second URL, and set this URL as the value of token_endpoint of the OpenId Provider Configuration Document.

2.2.3.1.2. Request parameters

AuthEndpoint:
The endpoint to request an access token after discovering the value by performing the authentication flow.

AuthSubpath:
If a host system uses a relative subpath dedicated to serving an OpenId Provider Configuration Document and creating an access token, then the requesting data recipient MUST use this subpath.

AuthHostname
The requesting data recipient MUST use the domain name of the host system dedicated to serving an OpenId Provider Configuration Document and creating an access token.

BasicAuth
See RFC6749 Section 4.4.2

ContentLength
The length of the body. See RFC9112.

AuthBody
See RFC6749 Section 4.4
The client MUST additionally request the following parameters.
Grant type: Mandatory. The value MUST be “client_credentials”.
Scope: Optional. The scope of access requirements is explained in RFC6749 Section 3.3.

2.2.3.1.3. Response syntax

HTTP/1.1 AuthStatusCode OK
content-type: application/json
content-length: ContentLength

AuthResponseBody

2.2.3.1.4. Response parameters

AuthStatusCode
A HTTP response code conforming to RFC6749 Section 4.4 and Section 5.

AuthResponseBody
A JSON Object conforming to either RFC6749 Section 4.4 in case of successful authentication (containing an access token), or RFC6749 Section 5.2 otherwise.

2.2.3.1.5. Example

Example of a successful response

```
{
   "access_token": "...",
   "token_type": "bearer"
}
```

Example of an error response
Example HTTP call, for instance generated because username or password did not match:

```
HTTP/1.1 401 Unauthorized
date: Mon, 23 Oct 2023 19:33:16 GMT
content-type: application/json
{
   "error": "invalid_client",
   "error_description": "Authentication failed"
}
```

For further details, for instance on the list of specified values of property error, consult RFC6749 Section 5.2.

2.2.3.2. ListFootprints

ListFootprints references Action ListFootprints in the Pathfinder Network API with two differences: changing the Request syntax path and adding the Green x Digital Consortium's own data model to the Response parameters.

Lists product footprints with pagination and optional filtering.

Host systems SHOULD implement an access management system and only return the product footprints for which the data owner granted access to the requesting data recipient.

2.2.3.2.1. Filtering

Filtering can be requested by a data recipient by setting a conforming Filter request parameter.
[Note]
The filter statement syntax is described at the Filter request parameter.

Support for filtering by a host system is optional such that:

1) if a host system does not implement Filtering, it MUST process the request as if no Filter was provided
2) If a host system implements Filtering, it can process the filter statement on a best-effort basis:
    a. it can ignore the filter statement or parts of the filter statement, or
    b. it can refuse to process the specific filter statement if it does not support the filter statement or parts of it. In this case, it MUST return an error response with code NotImplemented.

2.2.3.2.2. Pagination

Host systems MUST implement pagination server-side such that:

1) The host system MAY return less ProductFootprints than requested through the Limit request parameter
2) The host system MUST return a Link header if there are additional ProductFootprints ready to be retrieved, such that
3) The Link header conforms to [RFC8288]
4) The value of the rel parameter is equal to next
5) The target IRI (RFC8288, section 3.1) of the Link header is absolute
6) The value of host of the target IRI is equal to the value of the host request header from the original ListFootprints HTTP request

The target IRI from a pagination link header is called a pagination link.

Upon a host system returning a pagination link:

1) A data recipient can call the pagination link more than once.
2) Upon each call, the host system
    a. MUST return the same set of Product Footprints upon successful authentication (i.e. a Bearer token authentication as defined in § 6.3 Authentication Flow).
    b. MUST not return more product footprints than requested in case Limit was defined by a data recipient.
    c. MUST return a Link header conforming with the previous description in case there are additional ProductFootprints available.
3) If a response contains a second pagination link and the data recipient upon calling the second pagination link, the previous pagination link MAY no longer work, i.e., data recipients MUST not assume that previous pagination links continue to return results after advancing in the pagination process.
4) A pagination link MUST be valid for at least 180 seconds after creation.
5) A data recipient SHOULD retry calling the pagination link after the server returned an error.
6) A data recipient SHOULD use a randomized exponential back-off strategy when retrying.

2.2.3.2.3. Request syntax

```
GET Subpath/JEITA/0/footprints?Filter HTTP/1.1
Host: Hostname
authorization: Bearer BearerToken
```

2.2.3.2.4. Request parameters

Subpath: If a host system uses a relative subpath, then the requesting data recipient MUST prepend this subpath.

Hostname: The requesting data recipient MUST use the domain name of the host system.

BearerToken: The requesting data recipient MUST use as a BearerToken the value of property access_token of a response body from action Authenticate.

Filter: Filter is an optional request parameter. If defined, the value of the HTTP request parameter MUST conform to the $filter syntax as defined by the ODatav4 specification.

Additionally, the $filter statement MUST only include the following operators and properties: Logical operators eq, lt, le, gt, ge on properties created, updated, productCategoryCpc, geographyCountry, referencePeriodStart, referencePeriodEnd.

- Logical operator and
- Lambda operator any on collection-valued properties companyIds, productIds. The expression argument of the operator MUST only include the eq operator.

Limit: Limit is an OPTIONAL request parameter. If defined, the name of the HTTP request parameter MUST be limit and the value MUST be a positive integer.

2.2.3.2.5. Response syntax

HTTP/1.1 HttpStatusCode OK
content-type: application/json
content-length: ContentLength
Body

2.2.3.2.6. Response parameters

HttpStatusCode:
If the host system returns a list of product footprints, the HttpStatusCode MUST be either 200 or 202:

- HttpStatusCode 200 indicates the returned list is the complete result of the given query.
- HttpStatusCode 202 indicates the returned list is an incomplete result of the given query. The host system MAY return this HttpStatusCode if it principally decides that it's able to obtain the remaining data in the future. This HttpStatusCode MUST not be returned if the request parameter Filter is not defined. The data recipient MAY continue to send the same request with exponential-backoff until it receives the complete result, indicated by HttpStatusCode 200.

If the host system responds with an error response, the HttpStatusCode MUST match the HTTP Status Code of the respective error response code.

If the host system does not return the list of ProductFootprints, it MUST return an error HTTP Status Code (4xx, 5xx).

ContentLength:
Length of the body. See RFC 2616.

Body:
If the host system accepts the access token, the body MUST be a JSON object with property data with value the list of ProductFootprints. The list MUST be encoded as a JSON array. If the list is empty, the host system MUST return an empty JSON array.

The host system MUST return the latest version of each footprint and MAY return previous versions. Among the footprints with identical id values, the one with the maximum version value is called the latest version and the rest are called the previous versions. If the request parameter Filter is defined, the specified expression SHOULD be evaluated for each ProductFootprint in the collection as described in OData v4 specification, and only ProductFootprints where the expression evaluates

to true SHOULD be included in the response. ProductFootprints for which the expression evaluates to false or which are not made available for the data recipient SHOULD be omitted from the list returned in the response.

If the host system does not accept the access token, the body MUST be an error response with code AccessDenied.

If the host system does not accept the access token because it expired, the body SHOULD be an error response with code TokenExpired.

In all other cases, for instance in case of a malformed value of the header authorization, the body SHOULD be an error response with code BadRequest.

2.2.3.2.7. Example

Request

```
GET example/0/footprints/497f6eca-6276-4993-bfeb-53cbbbba6f08 HTTP/1.1
host: a.example.com
authorization: Bearer PIYOpiyoPIYOpiyoPIYOpiyo
```

Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 300

{"data":{
        "id":“497f6eca-6276-4993-bfeb-53cbbbba6f08",
        "specVersion":"1.0.0",
        "version":0,
        ...,
        "pcf":{
                "declaredUnit":"kilogram",
                "unitaryProductAmount":0.00204,
                "fossilGhgEmissions":"8.6",
                ...,
                "gateTogate":{
                        ...
                },
                ...
        }
}]}
```

3.2.3.3 GetFootprint

GetFootprint references Action GetFootprint in the Pathfinder Network API with two differences: changing the Request syntax path and adding the Green x Digital Consortium’s own data model to the Response parameters.

Retrieves product footprints.

Host systems SHOULD implement an access management system and only return the product footprints for which the data owner granted access to the requesting data recipient.

3.2.3.3.1. Request syntax

```
GET Subpath/JEITA/2/footprints/PfId HTTP/1.1
Host: Hostname
authorization: Bearer BearerToken
```

3.2.3.3.2. Request parameters

Subpath:
If a host system uses a relative subpath, then the requesting data recipient MUST prepend this subpath.

Hostname
The requesting data recipient MUST use the domain name of the host system.

BearerToken: The requesting data recipient MUST use as a BearerToken the value of property access_token of a response body from action Authenticate.

PfId: The requesting data recipient MUST use as PfId the id of a product footprint it intends to retrieve.

3.2.3.3.3. Response syntax

```
HTTP/1.1 HttpStatusCode OK
content-type: application/json
content-length: ContentLength
Body
```

3.2.3.3.4. Response parameters

HttpStatusCode:
If the host system returns a list of product footprints, the HttpStatusCode MUST be 200. If the host system responds with an error response, the HttpStatusCode MUST match the HTTP Status Code of the respective error response code. If the host system does not return the list of ProductFootprints, it MUST return an error HTTP Status Code (4xx, 5xx).

ContentLength:
Length of the body. See RFC2616.

Body:
If the host system accepts the access token and allows the requesting data recipient to access the requested product footprint, the body MUST be a JSON object with property data. The value of property data MUST be the product footprint with footprint identifier GetPfId. If there were changes to the requested product footprint with identifier GetPfId, the host system SHOULD return the latest product footprint identified with identifier GetPfId and the maximum value of property

version.

If the host system accepts the access token, the body MUST be a JSON object with property data with value the list of ProductFootprints. The list MUST be encoded as a JSON array. If the list is empty, the host system MUST return an empty JSON array. If there were changes to the requested product footprint with identifier GetPfId, the host system SHOULD return the latest product footprint identified with identifier GetPfId and the maximum value of property version.

[Note]
If a host system implements the life cycle rules, then the “latest” version of the requested product footprint is the one with the maximum value of version with id equal to GetPfId.

If the host system does not accept the access token, the body MUST be an error response with code AccessDenied.

If the host system does not accept the access token because it expired, the body SHOULD be an error response with code TokenExpired.

The host system MAY return an error response with code NoSuchFootprint.

In all other cases, for instance in case of a malformed value of the header authorization, the body SHOULD be an error response with code BadRequest.

3.2.3.3.5. Example

Request

```
GET example/2/footprints/497f6eca-6276-4993-bfeb-53cbbbba6f08 HTTP/1.1
host: a.example.com
authorization: Bearer PIYOpiyoPIYOpiyoPIYOpiyo
```

Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 300

{"data":{
        "id":“497f6eca-6276-4993-bfeb-53cbbbba6f08",
        "specVersion":"1.0.0",
        "version":0,
        ...,
        "pcf":{
                "declaredUnit":"kilogram",
                "unitaryProductAmount":0.00204,
                "fossilGhgEmissions":"8.6",
                ...,
                "gateTogate":{
                        ...
                },
                ...
```

```
        }
    }}
```

3.2.3.4. ActionEvents

ActionEvents references Action Events in the Pathfinder Network API with two differences: changing the Request syntax path and adding the Green x Digital Consortium's own data model to the Response parameters.

Action Events enables the exchange of event data between data owners and data recipients.

The Action Events endpoint is specified for the following use cases:

1. enabling a data owner to notify a data recipient on updates to 1 or more Product Footprints (see 3.2.3.4.5 Notification of data recipients on Product Footprint updates)
2. enabling a data recipient to request product footprints from a data owner by sending an event to the data owner's Action Events endpoint (see 3.2.3.4.6 Asynchronous request and retrieval of Product Footprints).

A host system SHOULD only accept events after authentication.

The Action Events endpoint accepts CloudEvent events (see [CE]) encoded in "Structured Content Mode" (see [CE-Structured-Content-Mode].

Support for Action Events is optional.

If a host system does not implement the Action Events endpoint,

1. it SHOULD respond with a conforming error response and HTTP error response code.
2. it SHOULD respond to authenticated Action Events calls with an error response with code NotImplemented.
3. it MUST respond with an error HTTP Status Code (4xx, 5xx).

3.2.3.4.1. Request syntax

```
POST Subpath/JEITA/2/events HTTP/1.1
Host: Hostname
authorization: Bearer BearerToken
content-type: application/cloudevents+json; charset=UTF-8

EventBody
```

3.2.3.4.2. Request parameters

Subpath:
If a host system uses a relative subpath, then the requesting data recipient MUST prepend this subpath.

Hostname
The requesting data recipient MUST use the domain name of the host system.

BearerToken: The requesting data recipient MUST use as a BearerToken the value of property access_token of a response body from action Authenticate.

EventBody
The EventBody MUST be:

1. a CloudEvents event (see [CE])
2. encoded as a JSON object as defined in [CE-JSON]
3. using "Structured Content Mode" (see [CE-Structured-Content-Mode]).

Further details on the EventBody syntax and semantics are given in 3.2.3.4.5. Notification of data recipients on Product Footprint updates and 3.2.3.4.6. Asynchronous request and retrieval of Product Footprints.

3.2.3.4.3. Response syntax

HTTP/1.1 HttpStatusCode OK
content-length: 0

3.2.3.4.4. Response parameters

HttpStatusCode:
The host system accepting the event MUST respond with the HttpStatusCode 200. The host system upon not accepting the event SHOULD respond with an error response:

ContentLength:
Length of the body. See RFC 2616.
The host system upon accepting the event MUST respond with an empty body.

3.2.3.4.5. Notification of data recipients on Product Footprint updates

A data owner can notify a data recipient about changes to 1 or more product footprints by sending a PF Update Event to the data recipient’s Action Events endpoint.

A data recipient upon receiving such an PF Update Event can retrieve the product footprints through the Action GetFootprint.

Accordingly, the data owner of the host system sending the event MUST make the referenced Product Footprints available to the data recipient notified through the PF Update Event.

The PF Update Event is defined as a JSON-encoded CloudEvent event with the following syntax:

```
{
   "type": "org.wbcsd.pathfinder.ProductFootprint.Published.v1",
   "specversion": "1.0",
   "id": "EventId",
   "source": "//EventHostname/EventSubpath",
   "time": "2022-05-31T17:31:00Z",
   "data": {
```

```
        "pfIds": PfIds
      }
    }
```

EventId
A unique identifier for the event set by the host system sending the event. The EventId MUST be a string (see [CE-JSON]).

PfIds
A list of product footprint that have been updated. The PfIds MUST be the non-empty list of id values of the updated Product Footprints, encoded as a JSON array.

EventHostname
The Hostname of the host system sending the event.

EventSubpath
The handler of the host system sending the event.

3.2.3.4.6. Asynchronous request and retrieval of Product Footprints

A data recipient can request a data owner to send a product footprint by sending a PF Request Event to the data owner's Action Events endpoint.

A data owner upon receiving a PF Request Event can then decide how to process the request

1. by sending a PF Response Event to the data recipient's Action Events endpoint
2. by sending a PF Response Error Event to the data recipient's Action Events endpoint to notify the data recipient that the request cannot be processed, or
3. by not sending any event to the data recipient's Action Events endpoint.

If a data owner accepted a PF Request Event, the host system MUST send the response back to the host system referenced in source of the PF Request Event. The host system of the data owner MUST validate the value of source before sending the response. If the host system of the original requestor (the data recipient) is not available or does not accept the response with a HTTP success code (2xx), the data owner's host system SHOULD retry sending the response event using exponential backoff. A host system SHOULD NOT retry sending a response event for more than 3 days.

3.2.3.4.6.1. PF Request Event syntax

The PF Request Event is defined as a CloudEvent event sent from a data recipient to a data owner. The event contains a Product Footprint fragment to describe to the data owner the requested product footprints.

The PF Request Event is defined as a JSON-encoded CloudEvent event with the following syntax:

```
{
  "type": "org.wbcsd.pathfinder.ProductFootprintRequest.Created.v1",
  "specversion": "1.0",
```

```
    "id": "EventId",
    "source": "//EventHostname/EventSubpath",
    "time": "2022-05-31T17:31:00Z",
    "data": {
      "pf": ProductFootprintFragment,
      "comment": PFRequestComment
    }
  }
```

EventId
A unique identifier for the event set by the host system sending the event. The EventId MUST be a string (see [CE-JSON]).

EventHostname
The Hostname of the host system sending the event.

EventSubpath
The handler of the host system sending the event.

ProductFootprintFragment
A JSON object which references a subset of ProductFootprint properties, including nested properties.

PFRequestComment
The optional comment by the data recipient to the data owner about the request. If defined, the PFRequestComment MUST be encoded as a JSON string.

The property data.comment of a PF Request Event is optional.

3.2.3.4.6.2. PF Response Event syntax

The PF Response Event is defined as a CloudEvent event sent from a data owner to a data recipient after having received a PF Request Event from a data recipient and upon successfully fulfilling the request.

The PF Response Event is defined as a JSON-encoded CloudEvent event with the following syntax:

```
{
  "type": "org.wbcsd.pathfinder.ProductFootprintRequest.Fulfilled.v1",
  "specversion": "1.0",
  "id": "EventId",
  "source": "//EventHostname/EventSubpath",
  "data": {
    "requestEventId": "ReqEventId",
    "pfs": Pfs
  }
}
```

ReqEventId
The EventId of the PF Request Event that the PF Response Event is responding to. The ReqEventId MUST be a string (see [CE-JSON]).

Pfs
The list of product footprints that have been requested with the PF Request Event and that are accessible to the data recipient, encoded as an array of ProductFootprint in JSON.

Otherwise, the value of Pfs MUST be the empty JSON array.

3.2.3.4.6.3. PF Response Error Event syntax

The PF Response Error Event is defined as a CloudEvent event sent from a data owner to a data recipient after having received a PF Request Event from a data recipient and upon not successfully fulfilling the request.

The PF Response Event is defined as a JSON-encoded CloudEvent event with the following syntax:

```
{
   "type": "org.wbcsd.pathfinder.ProductFootprintRequest.Rejected.v1",
   "specversion": "1.0",
   "id": "EventId",
   "source": "...",
   "data": {
      "requestEventId": "ReqEventId",
      "error": ReqErrorResponse
   }
}
```

ReqErrorResponse
The error response that the data owner is sending to the data recipient to notify the data recipient that the request cannot be processed.

The value of ReqErrorResponse MUST be an error response.

3.2.3.5. Error responses

The actions Action GetFootprint, Action ListFootprints, and Action Events specify general error response handling.

Error responses are specified in detail such that data recipients can understand the cause of the error, and so that potentially host systems can react on and resolve errors automatically.

NOTE: Action Authenticate specifies its own error responses (see § 6.5.2 Response Syntax).

Whenever a host system returns an error response, it MUST send a HTTP response such that

- the HTTP Status Code equals the HTTP Status Code defined for the respective error response code (see Error Codes Table)

- with content type set to application/json, and
- with response body the error response

A error response is a JSON object with the following properties:

- code: a error response code encoded as a String
- message: a error message encoded as a String

A error response code is a value from column Error Response Code from 3.2.3.5.1. List of error codes.

Example: AccessDenied error response

```
{
"message": "Access Denied",
"code": "AccessDenied"
}
```

A host system MAY return error messages different from the list below, for instance localized values depending on a data recipient.

3.2.3.5.1. List of error codes

A error message is a human-readable error description. Example values are in column Example Message in the table below. A host system MAY return error messages different from the table below, for instance localized values depending on a data recipient.

Error Response Code	Example Message	HTTP Status Code
AccessDenied	Access denied	403
BadRequest	Bad Request	400
NoSuchFootprint	The specified footprint does not exist.	404
NotImplemented	The specified Action or header you provided implies functionality that is not implemented	400
TokenExpired	The specified access token has expired	401
InternalError	An internal or unexpected error has occurred	500

3.2.3.5.2. Error processing by a data recipient

A requesting data recipient MUST use the code property and potentially also the HTTP Status Code to differentiate between the different errors.

3.3. References

[CE]
Cloud Events Specification. LS. URL: https://github.com/cloudevents/spec

[CE-Structured-Content-Mode]
HTTP Protocol Binding for CloudEvents - Version 1.0.2. LS. URL: https://github.com/cloudevents/spec/blob/v1.0.2/cloudevents/bindings/http-protocol-binding.md#32-structured-content-mode

[CE-JSON]
JSON Event Format for CloudEvents - Version 1.0.2. LS. URL: https://github.com/cloudevents/spec/blob/v1.0.2/cloudevents/formats/json-format.md