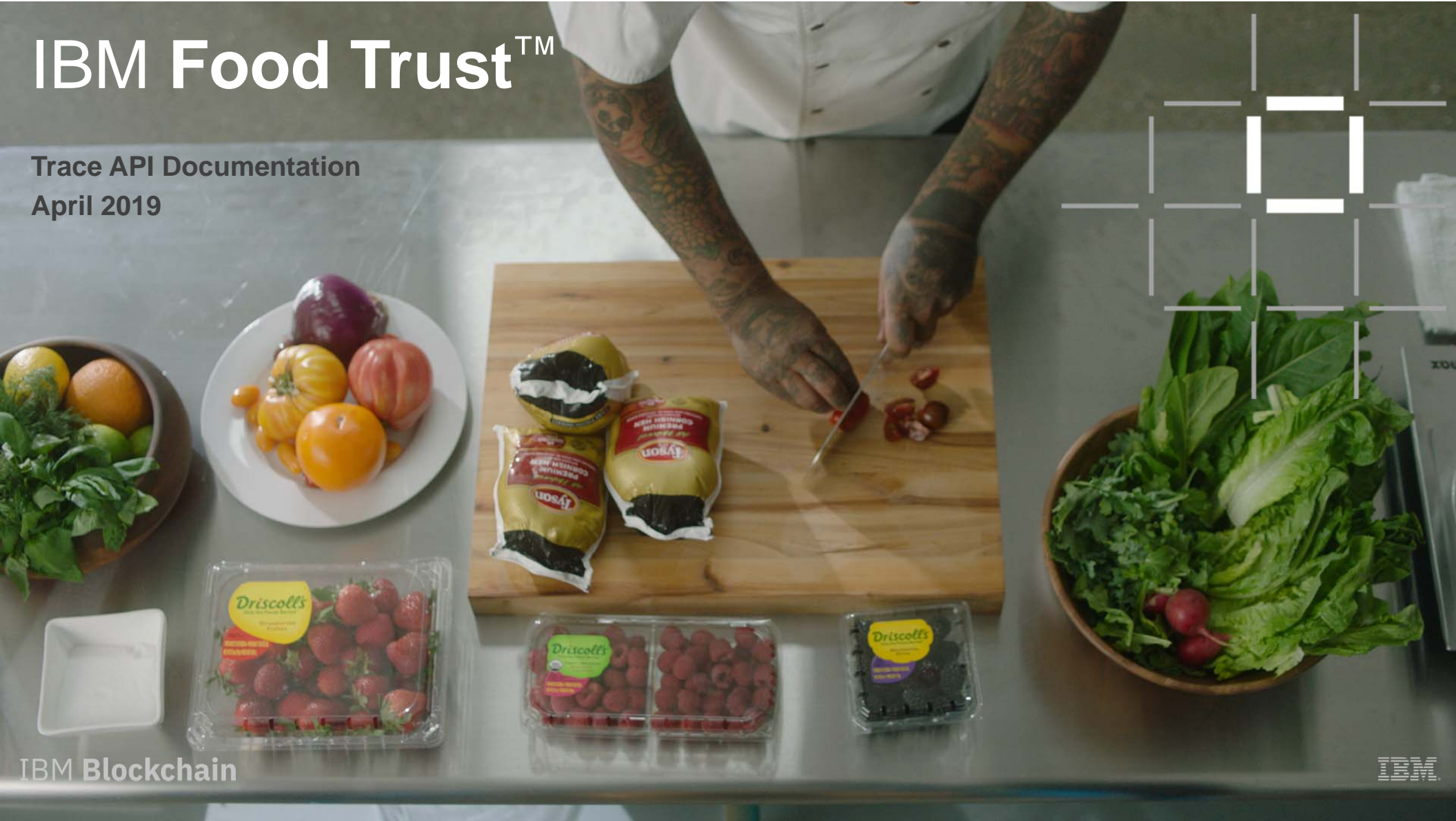


IBM Food Trust™

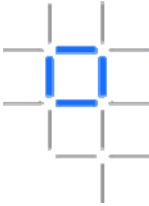
Trace API Documentation
April 2019



IBM Blockchain

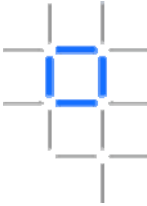


Table of Contents



- **Overview**
- Example scenario
- Obtaining scenario details from additional endpoints
 - Events
 - Products
 - Locations
 - Lots and Serials

Overview



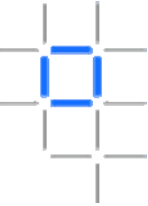
The Trace API:

- allows users to construct a view of a lot of product as it flows through a supply-chain process.
- broken into separate endpoints to allow for maximum flexibility to meet the needs of a wide variety of customers and situations

Note: Please visit the Swagger page for more detailed information about the endpoints and their various parameters.

Endpoint	Method	Description
Trace: food.ibm.com/ift/api/trace/v2/epcs/{id}/trace	GET	Returns all of the EPCs required to trace a Lot/Serial through IBM Food Trust
Events: food.ibm.com/ift/api/trace/v2/events	GET	Returns event details
Products: food.ibm.com/ift/api/trace/v2/products	GET	Returns master item details
Locations: food.ibm.com/ift/api/trace/v2/locations	GET	Returns master location details
Payloads: food.ibm.com/ift/api/trace/v2/payloads	GET	Returns payload details
Lots and Serials: food.ibm.com/ift/api/trace/v2/lots_and_serials	GET	Returns details about lots and serials
Consumer: food.ibm.com/ift/api/trace/v2/epcs/{id}/trace/consumer	GET	Composite endpoint for use by client's applications that includes all of the trace information. Available for client selected products/lots.

Consumer Endpoint

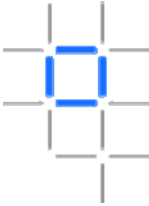


What is it?

- Endpoint that provides a composite response of trace data relevant for a consumer facing traceability presentation
 - EPCs
 - Events
 - Products
 - Locations
 - Payloads
 - Lots and Serials
- Additionally supports low-latency response time when configured to pre-compute product traces for specific GTINs/IBM Food Trust Identifiers

How does “low-latency response” work?

1. Client selects the products/lots that are the most important for their business needs
2. IBM Food Trust ensures that the selected data is always present in the cache for fast trace response



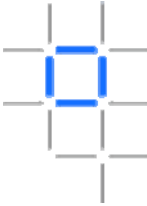
Trace Endpoint Details

/epcs/{id}/trace

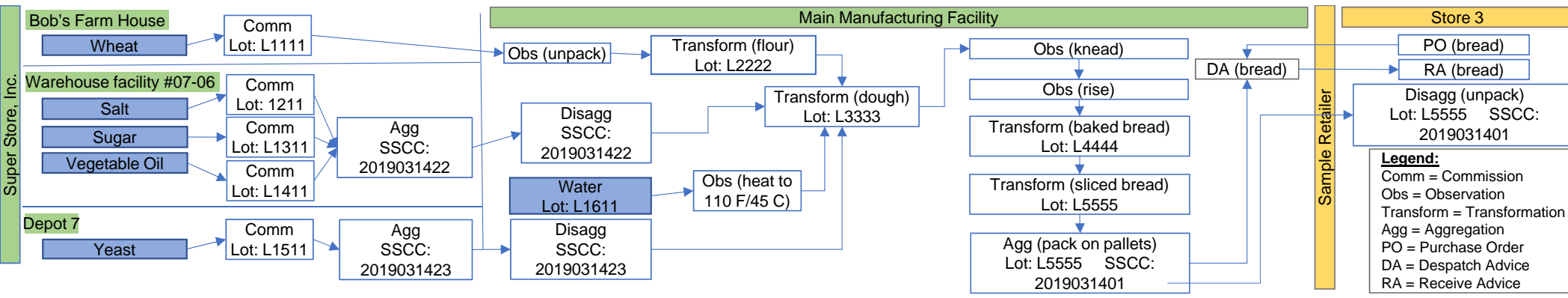
Response element	Event Type						Notes
	Commission	Decommission	Observation	Aggregation	Disaggregation	Transformation	
events[]	X	X	X	X*	X*	X*	May contain asset_ids for all events associated with the EPC that was traced.
input_epcs[]	X	X	X			X	Downstream EPCs. The EPC(s) at this hop level have the previous hop level EPC as an input.
output_epcs[]	X	X	X			X	Upstream EPCs. The EPC(s) at this hop level have the previous hop level EPC as an output.
parent_epcs[]	X	X	X	X	X		EPCs that wholly contained the traced EPC (e.g. Lot(s) that were part of an aggregation event involving the previous hop level EPC)
child_epcs[]	X	X	X	X	X		EPCs that are part of the traced EPC (e.g. Lot(s) that were created after aggregating EPC(s))

* The events[] array for the root element will only not contain these elements because they will be listed in their respective input_epcs/output_epcs and parent_epcs/child_epcs array elements.

Table of Contents



- Overview
- **Example scenario**
- Obtaining scenario details from additional endpoints
 - Events
 - Products
 - Locations
 - Lots and Serials



Example Scenario: A fictional shipment of sliced bread

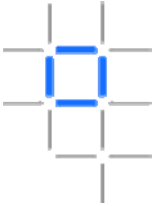
Overview:

Raw ingredients come from multiple locations (farm, warehouse, Depot) and are combined in a manufacturing facility which turns it into sliced bread. Pallets of these loaves of bread are sent to a retailer for distribution to a store.

Scenario Organizations:

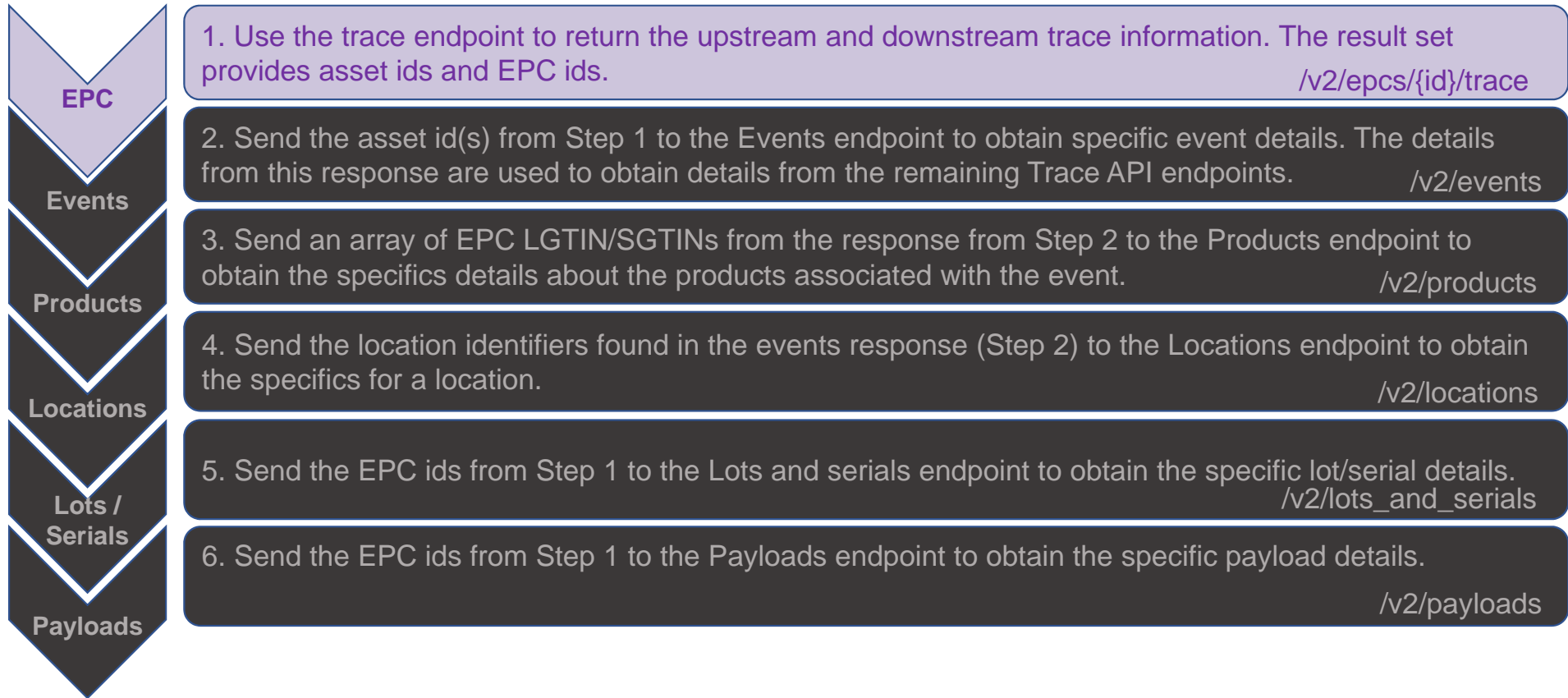
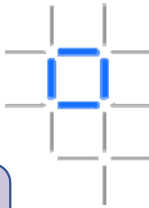
- Sample Retailer (owns/runs multiple retail stores)
 - Store 3
- Super Store, Inc. (owns it's own farm and multiple facilities)
 - Bob's Farm House (wheat farmer)
 - Warehouse Facility #07-06 (stores multiple individual ingredients used in manufacturing products: salt, sugar, vegetable oil)
 - Depot 7 (supplies yeast)
 - Main Manufacturing Facility (transforms raw ingredients into finished products and ships them)

Table of Contents



- Overview
- Example scenario
- **Obtaining scenario details from additional endpoints**
 - Events
 - Products
 - Locations
 - Lots and Serials

Leveraging the Trace API endpoints to obtain the sliced bread information

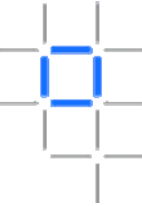




EPC

1. Use the trace endpoint to return the upstream and downstream trace information. The result set provides asset ids and EPC ids.

/v2/epcs/{id}/trace



Example 1: Partial trace from the “Retailer”

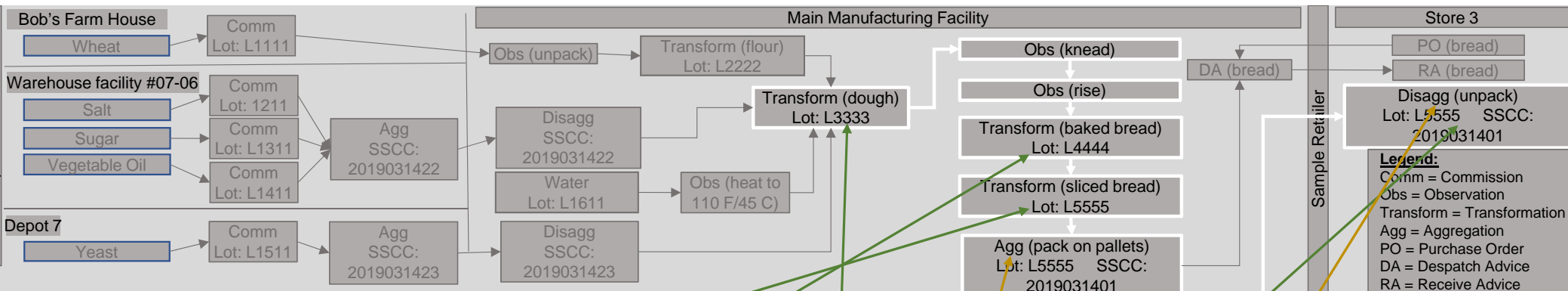
CURL:

```
curl -X GET 'https://food.ibm.com/ift/api/trace/v2/epcs/urn:ibm:ift:product:lot:class:0002662.Ifyl.L5555/trace?depth=2' -H 'Authorization: Bearer ey...cw'
```

Notes:

- “depth” parameter set to “2” for display purposes
- asset_id(s) have been truncated for space purposes
- The trace output will show only “input_epcs” because the API call is tracing on the furthest downstream lot; all other EPCs will be inputs to Lot L5555.

Super Store, Inc.



Legend:
 Comm = Commission
 Obs = Observation
 Transform = Transformation
 Agg = Aggregation
 PO = Purchase Order
 DA = Despatch Advice
 RA = Receive Advice

```

{
  "epc_id": "urn:ibm:ift:product:lot:class:0002662.Ifyl.L5555",
  "input_epcs": [ {
    "epc_id": "urn:ibm:ift:product:lot:class:0002662.iohF.L4444",
    "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ],
    "input_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.jPGc.L3333",
      "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." },
        { "asset_id": "urn:ibm:provenance:asset:event:observation:..." },
        { "asset_id": "urn:ibm:provenance:asset:event:observation:..." }
      ],
      "input_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.bxMa.L2222",
        "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ]
      }, { "epc_id": "urn:ibm:ift:product:lot:class:0002662.NBWh.L1211",
        "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ]
      }, { "epc_id": "urn:ibm:ift:product:lot:class:0013265.mPWq.L1311",
        "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ]
      }, { "epc_id": "urn:ibm:ift:product:lot:class:0040097.aCqD.L1411",
        "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ]
      }, { "epc_id": "urn:ibm:ift:product:lot:class:0065466.0aYv.L1511",
        "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ]
      }, { "epc_id": "urn:ibm:ift:product:lot:class:0883655000.wthX.L1611",
        "events": [ { "asset_id": "urn:ibm:provenance:asset:event:transformation:..." } ]
      }
    ]
  } ]
}, {
  "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031401",
    "events": [ { "asset_id": "urn:ibm:provenance:asset:event:aggregation:..." },
      { "asset_id": "urn:ibm:provenance:asset:event:disaggregation:..." } ]
  } ]
}

```

Lot L5555 had an input of L4444 which was a Transformation. Since Lot L4444 is in the input_epcs array, it is an input to the previous hop.

Lot L3333 has three events, a Transformation and two Observation events. Since Lot L3333 is in the input_epcs array, it is an input to the previous hop, Lot L4444.

Because depth=2, Lot L3333 has six (6) input_epcs which are the ingredients to the bread dough, but we won't see any additional details at this point.

An additional call to the trace endpoint on Lot L3333 will yield more details.

This trace has one parent EPC which is the pallet that was shipped from the Main Manufacturing Facility (Agg event) to Store 3 (DisAgg event).



EPC

1. Use the trace endpoint to return the upstream and downstream trace information. The result set provides asset ids and EPC ids.

/v2/epcs/{id}/trace



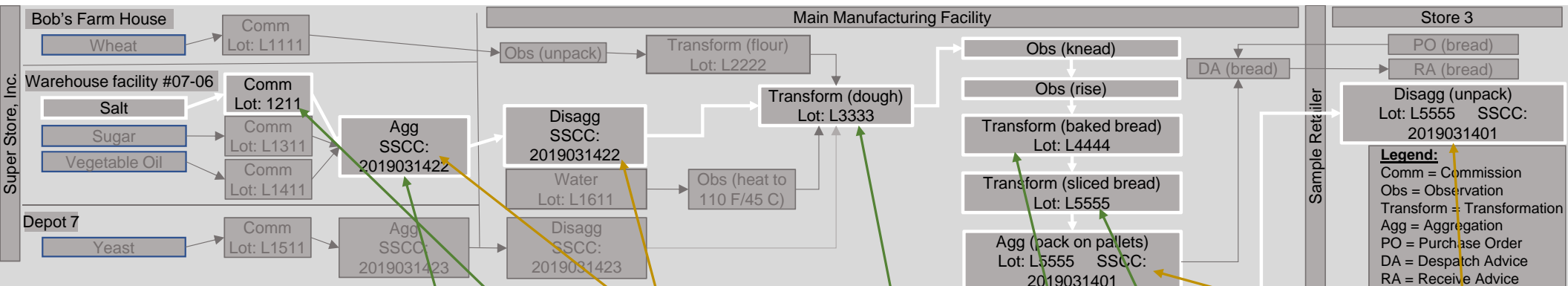
Example 2: Full trace from the “Supplier”

CURL:

```
curl -X GET 'https://food.ibm.com/ift/api/trace/v2/epcs/urn:ibm:ift:product:lot:class:0002662.NBWh.L1211/trace' -H 'Authorization: Bearer ey...cw'
```

Notes:

- asset_id(s) have been truncated for space purposes
- The trace output will show only “output_epcs” because the API call is tracing on the furthest upstream lot; all other EPCs will be outputs to Lot L1211.



```

{
  "epc_id": "urn:ibm:ift:product:lot:class:0002662.NBWh L1211",
  "events": [ { "asset_id": "...:event:commission:..." } ],
  "output_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.jpGc L3333",
    "events": [ { "asset_id": "...:event:transformation:..." },
      { "asset_id": "...:event:observation:..." },
      { "asset_id": "...:event:observation:..." }
    ],
    "output_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.iohF L4444",
      "events": [ { "asset_id": "...:event:transformation:..." } ],
      "output_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.Ifyl L5555",
        "events": [ { "asset_id": "...:event:transformation:..." } ],
        "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031401",
          "events": [ { "asset_id": "...:event:aggregation:..." },
            { "asset_id": "...:event:disaggregation:..." }
          ]
        }
      ]
    }
    ]
  },
  "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031422",
    "events": [ { "asset_id": "...:event:aggregation:..." },
      { "asset_id": "...:event:disaggregation:..." }
    ]
  }
]
}

```

Lot L1211 is associated with a Commission event and a single output EPC (Lot L3333)

Lot L3333 is associated with 3 events, the two Observations (knead and rise) and the Transformation whose output is baked bread.

event:aggregation:...

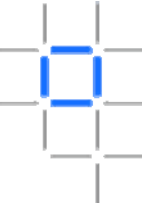
event:disaggregation:...



EPC

1. Use the trace endpoint to return the upstream and downstream trace information. The result set provides asset ids and EPC ids.

/v2/epcs/{id}/trace



Example 3: Full trace from the “Manufacturer”

CURL:

```
curl -X GET 'https://food.ibm.com/ift/api/trace/v2/epcs/urn:ibm:ift:product:lot:class:0002662.JPGc.L3333/trace' -H 'Authorization: Bearer ey...cw'
```

Notes:

- asset_id(s) have been truncated for space purposes

Lot L3333 "bread dough" is in the "middle" of the process, so it contains both input EPCs and output EPCs.

```
{
  "epc_id": "urn:ibm:ift:product:lot:class:0002662.jpGc.L3333",
  "events": [ { "asset_id": "...:event:observation:" },
    { "asset_id": "...:event:observation:" } ],
  "output_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.iohF.L4444",
    "events": [ { "asset_id": "...:event:transformation:" } ],
    "output_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.Ifyl.L5555",
      "events": [ { "asset_id": "...:event:transformation:" } ],
      "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:9999999.2019031401",
        "events": [ { "asset_id": "...:event:aggregation:" } ]
      } ]
    } ]
  } ],
  "input_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.bxMa.L2222",
    "events": [ { "asset_id": "...:event:transformation:" } ],
    "input_epcs": [ { "epc_id": "urn:ibm:ift:product:lot:class:999999999.KXEh.L1111",
      "events": [ { "asset_id": "...:event:transformation:" },
        { "asset_id": "...:event:commission:" } ]
    } ]
  }, { "epc_id": "urn:ibm:ift:product:lot:class:0002662.NBWh.L1211",
    "events": [ { "asset_id": "...:event:transformation:" },
      { "asset_id": "...:event:commission:" } ],
    "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031422",
      "events": [ { "asset_id": "...:event:aggregation:" },
        { "asset_id": "...:event:disaggregation:" } ]
    } ]
  }, { "epc_id": "urn:ibm:ift:product:lot:class:0013265.mPWq.L1311",
    "events": [ { "asset_id": "...:event:transformation:" },
      { "asset_id": "...:event:commission:" } ],
    "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031422",
      "events": [ { "asset_id": "...:event:aggregation:" },
        { "asset_id": "...:event:disaggregation:" } ]
    } ]
  }, { "epc_id": "urn:ibm:ift:product:lot:class:0040097.aCqD.L1411",
    "events": [ { "asset_id": "...:event:transformation:" },
      { "asset_id": "...:event:commission:" } ],
    "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031422",
      "events": [ { "asset_id": "...:event:aggregation:" },
        { "asset_id": "...:event:disaggregation:" } ]
    } ]
  }, { "epc_id": "urn:ibm:ift:product:lot:class:0065466.0aYv.L1511",
    "events": [ { "asset_id": "...:event:transformation:" },
      { "asset_id": "...:event:commission:" } ],
    "parent_epcs": [ { "epc_id": "urn:epc:id:sscc:0002662.2019031423",
      "events": [ { "asset_id": "...:event:disaggregation:" },
        { "asset_id": "...:event:aggregation:" } ]
    } ]
  }, { "epc_id": "urn:ibm:ift:product:lot:class:0883655000.wthX.L1611",
    "events": [ { "asset_id": "...:event:transformation:" },
      { "asset_id": "...:event:observation:" } ]
  } ]
}
```

EPCs and events that are downstream from this point. This includes the kneading, rising, baking, slicing and the packing of the bread.

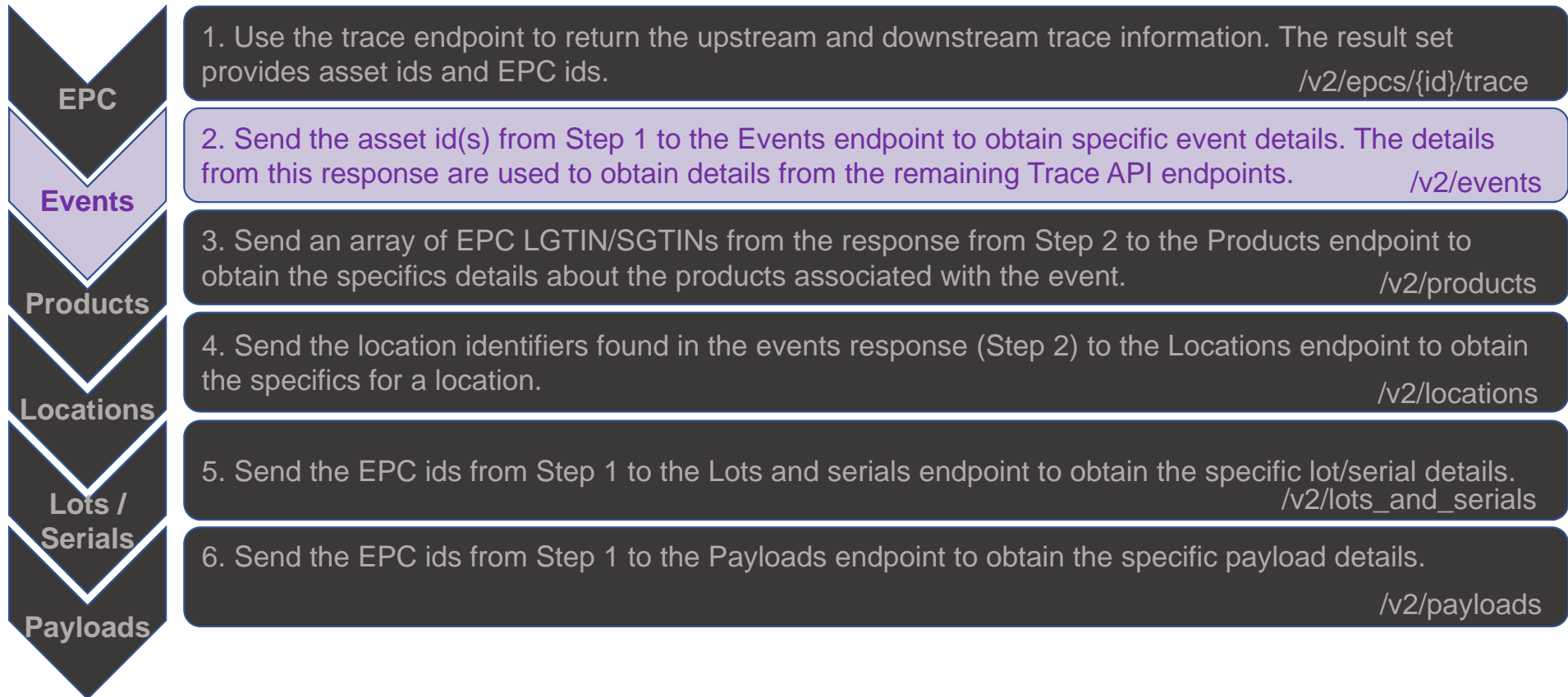
Each of the Input EPCs at this level specifies the Transformation event that was the event that took the lots and transformed them into the lot searched (L3333).

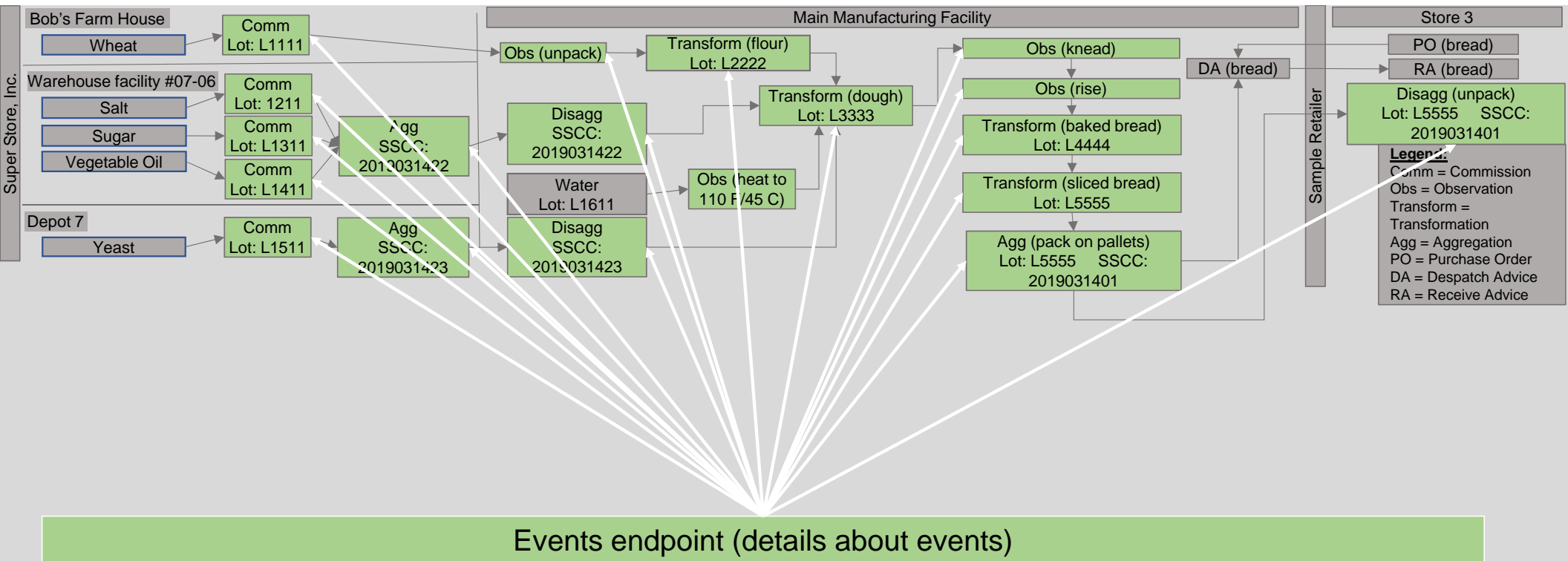
EPCs and events that are upstream from this point. This includes the base ingredients and the events that brought them to the traced Lot, L3333.

Parent EPC for a pallet of Salt, Sugar and Vegetable Oil. The Agg and DisAgg events appear for each of them representing the shipment from Warehouse facility #07-06 to the Manufacturing Plant.

Parent EPC for sending a pallet of Yeast. The Agg and DisAgg events appear for each of them representing the shipment from Depot #7 to the Manufacturing Plant.

Leveraging the Trace API endpoints to obtain the sliced bread information







2. Send the asset id(s) from Step 1 to the Events endpoint to obtain specific event details. The details from this response are used to obtain details from the remaining Trace API endpoints. /v2/events



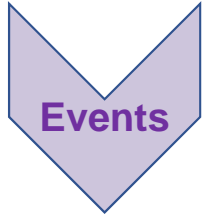
Commission Event: Lot L1211 (Salt)

```
curl -X GET 'https://food.ibm.com/ift/api/outbound/v2/events?asset_id[]=urn:ibm:provenance:asset:event:commission:...' -H 'Authorization: Bearer ey...Kw'
```

```
{ "events": [ {
  "id": "11112",
  "asset_id": "urn:ibm:provenance:asset:event:commission:...",
  "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",
  "event_type": "commission",
  "event_time": "2018-07-14T12:00:00.000Z",
  "read_point_location_id": "urn:ibm:ift:location:loc:0002662.DC9574",
  "read_point_sub_location_id": "urn:ibm:ift:location:loc:0002662.DC9574",
  "biz_location_id": "urn:ibm:ift:location:loc:0002662.DC9574",
  "biz_sub_location_id": "urn:ibm:ift:location:loc:0002662.DC9574",
  "epcs_ids": [ "urn:ibm:ift:product:lot:class:0002662.NBWh.L1211" ],
  "quantities": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.NBWh.L1211",
    "quantity": 20, "uom": "lb"
  } ],
  "child_quantities": [],
  "input_quantities": [],
  "output_quantities": [],
  "source_location_ids": [],
  "source_sub_location_ids": [],
  "destination_location_ids": [],
  "destination_sub_location_ids": [],
  "transaction_ids": []
} ]
}
```

Location and Read points of who and where the Salt was Commissioned.

Lot L1211 is 20 lbs of Salt



2. Send the asset id(s) from Step 1 to the Events endpoint to obtain specific event details. The details from this response are used to obtain details from the remaining Trace API endpoints. /v2/events



Transformation Event: Lot L3333

```
curl -X GET 'https://food.ibm.com/ift/api/outbound/v2/events?asset_id[]=urn:ibm:provenance:asset:event:transformation:...' -H 'Authorization: Bearer ey...Kw'
```

```
{ "events": [ {
  "id": "88888",
  "asset_id": "urn:ibm:provenance:asset:event:transformation:...",
  "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",
  "event_type": "transformation",
  "event_time": "2018-07-28T01:45:00.000Z",
  "read_point_location_id": "urn:ibm:ift:location:loc:0548138542.PFGs",
  "read_point_sub_location_id": "urn:ibm:ift:location:loc:0548138542.PFGs",
  "biz_location_id": "urn:ibm:ift:location:loc:0548138542.PFGs",
  "biz_sub_location_id": "urn:ibm:ift:location:loc:0548138542.PFGs",
  "epcs_ids": [ "urn:ibm:ift:product:lot:class:0002662.jPGc.L3333",
               "urn:ibm:ift:product:lot:class:0002662.iohF.L4444"
             ],
  "quantities": [],
  "child_quantities": [],
  "input_quantities": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.jPGc.L3333",
                          "quantity": 150, "uom": "lb"
                        } ],
  "output_quantities": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.iohF.L4444",
                           "quantity": 150, "uom": "lb"
                         } ],
  "source_location_ids": [],
  "source_sub_location_ids": [],
  "destination_location_ids": [],
  "destination_sub_location_ids": [],
  "transaction_ids": []
} ]
}
```

Location and Read points of who and where the dough was baked

Flattened list of all EPCs with this event

Input: Lot L3333 is 150 lbs of Bread Dough

Output: Lot L4444 is 150 lbs of Baked Bread

Events

2. Send the asset id(s) from Step 1 to the Events endpoint to obtain specific event details. The details from this response are used to obtain details from the remaining Trace API endpoints.

/v2/events



Aggregation Event: Lot L5555 (Sliced Bread)

```
curl -X GET 'https://food.ibm.com/ift/api/outbound/v2/events?asset_id[]=urn:ibm:provenance:asset:event:aggregation:...' -H 'Authorization: Bearer ey...Kw'
```

```
{ "events": [ { "id": "100000",
  "asset_id": "urn:ibm:provenance:asset:event:aggregation:...",
  "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",
  "event_type": "aggregation",
  "event_time": "2018-07-28T05:30:00.000Z",
  "biz_step": "urn:epcglobal:cbv:bizstep:packing",
  "disposition": "urn:epcglobal:cbv:disp:in_progress",
  "read_point_location_id": "urn:ibm:ift:location:loc:0002662.Ifyl",
  "read_point_sub_location_id": "urn:ibm:ift:location:loc:0002662.Ifyl",
  "biz_location_id": "urn:ibm:ift:location:loc:0002662.Ifyl",
  "biz_sub_location_id": "urn:ibm:ift:location:loc:0002662.Ifyl",
  "epcs_ids": [ "urn:ibm:ift:product:lot:class:0002662.Ifyl.L5555",
    "urn:epc:id:sscc:0002662.2019031401" ],
  "parent_epc_id": "urn:epc:id:sscc:0002662.2019031401",
  "quantities": [],
  "child_quantities": [ { "epc_id": "urn:ibm:ift:product:lot:class:0002662.Ifyl.L5555",
    "quantity": 150, "uom": "lb" } ],
  "input_quantities": [],
  "output_quantities": [],
  "source_location_ids": [ "urn:ibm:ift:location:loc:0002662.Ifyl" ],
  "source_sub_location_ids": [ "urn:ibm:ift:location:extension:loc:0002662.Ifyl.main_mfg_facility" ],
  "destination_location_ids": [ "urn:ibm:ift:location:loc:0548138542.PFGs" ],
  "destination_sub_location_ids": [ "urn:ibm:ift:location:extension:loc:0548138542.PFGs.store-3" ],
  "transaction_ids": [
    "urn:epcglobal:cbv:bt:0548138542.PFGs.P01",
    "urn:epcglobal:cbv:bt:0548138542.PFGs.P01-DA1"
  ]
} ]
}
```

Location and Read points of who and where the sliced bread was packed for shipment

Flattened list of all EPCs with this event

Parent EPC (SSCC pallet identifier)

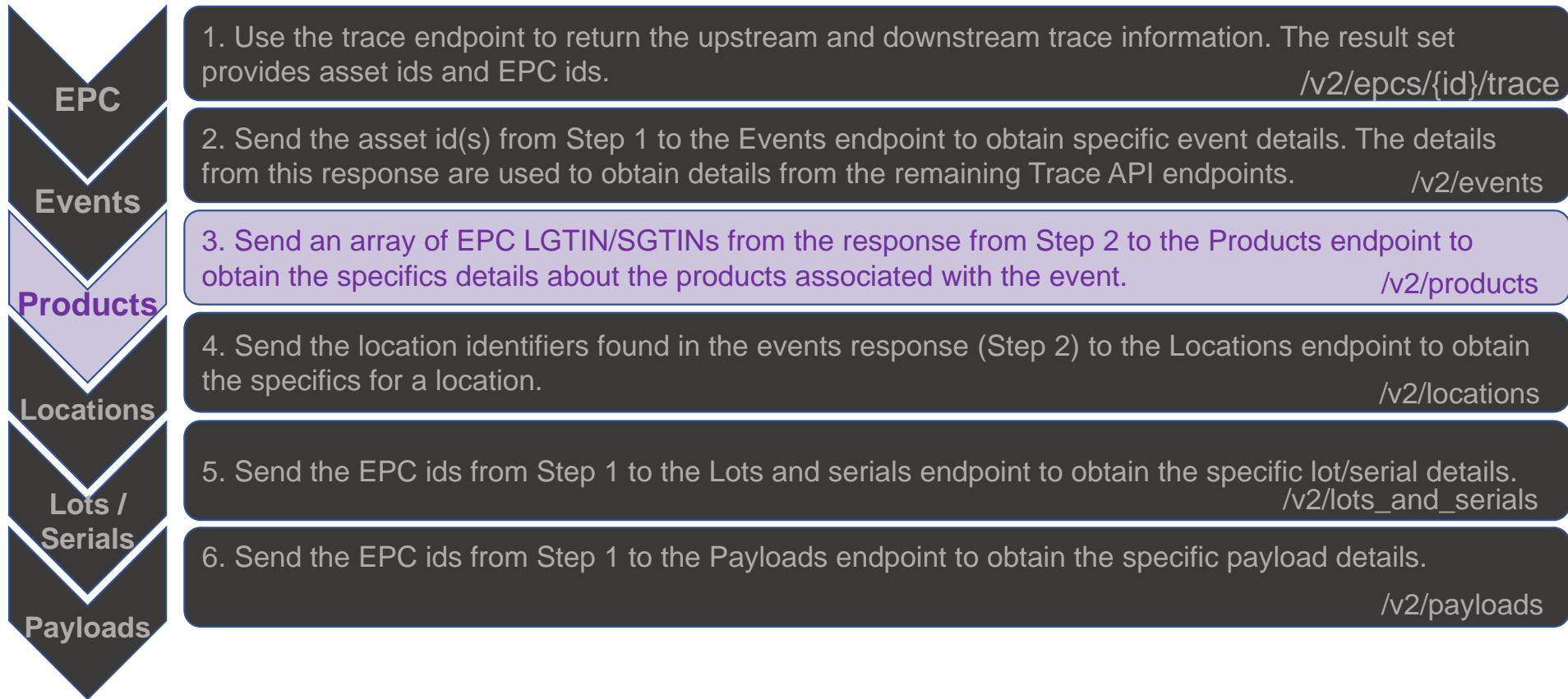
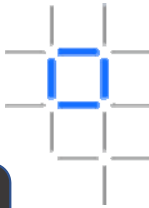
Details of how much Sliced Bread is on this pallet

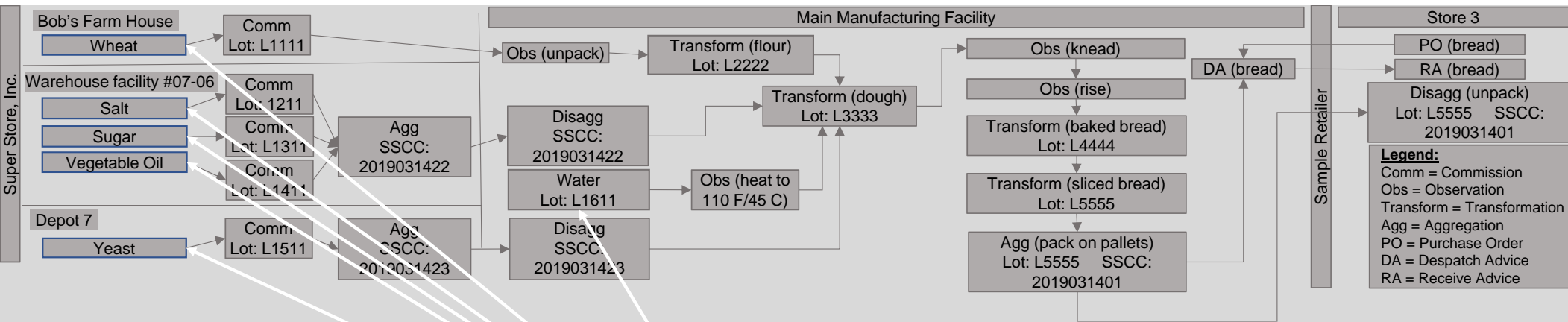
Primary company location

Discrete company location

Because this is a "last Aggregation" event signifying the transfer of product from one location to another, this array contains the list of the associated transactions with this event.

Leveraging the Trace API endpoints to obtain the sliced bread information





Products endpoint (details about products)



Products

3. Send an array of EPC LGTIN/SGTINs from the response from Step 2 to the Products endpoint to obtain the specifics details about the products associated with the event.

/v2/products



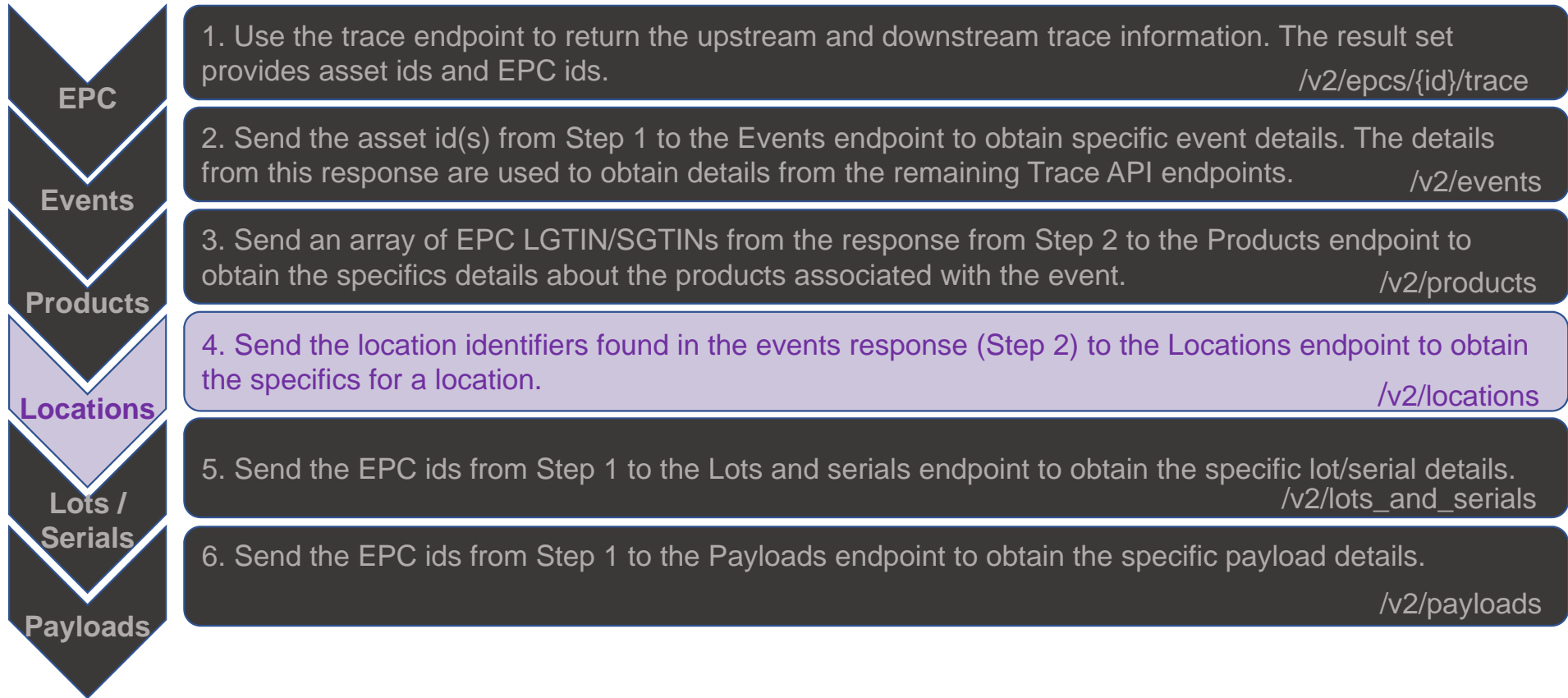
Obtain the product information for the wheat, salt and sugar ingredients:

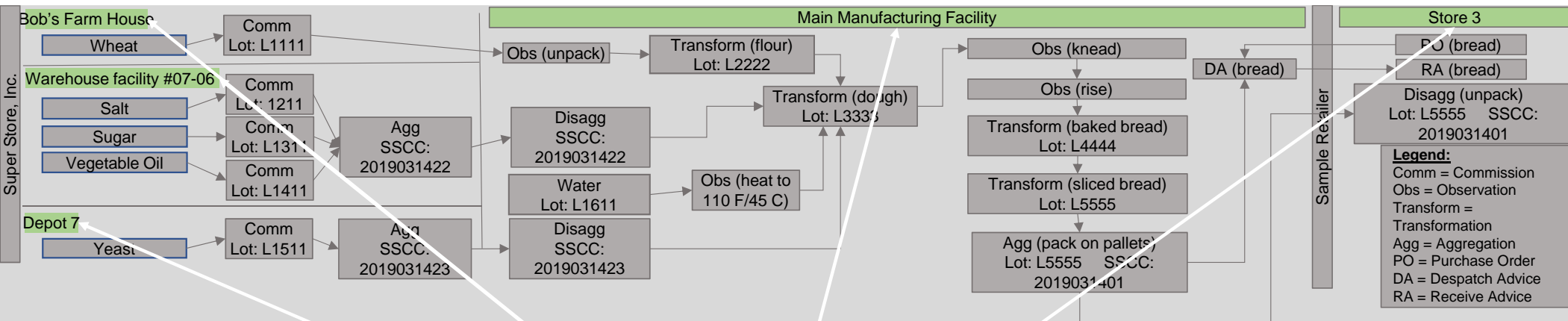
```
curl -X GET https://food.ibm.com/ift/api/trace/v2/products?product_id[]=urn:ibm:ift:product:class:0002662.mPWq&product_id[]=urn:ibm:ift:product:class:0002662.KXEh&product_id[]=urn:ibm:ift:product:class:0002662.NBWh -H 'Authorization: Bearer eyKw'
```

```
{ "products": [  
  {  
    "id": "urn:ibm:ift:product:class:0002662.KXEh",  
    "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",  
    "description": "Organic Wheat 10lbs",  
    "object_sku": "39fa06f02b9fa50c2a223c6efbad3fba"  
  }, {  
    "id": "urn:ibm:ift:product:class:0002662.NBWh",  
    "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",  
    "description": "Salt",  
    "object_sku": "b2827dcf84af03a34807eff8666c6927"  
  }, {  
    "id": "urn:ibm:ift:product:class:0002662.mPWq",  
    "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",  
    "description": "Sugar",  
    "object_sku": "07e8a3ee2dac9e38c7e1a7b1e919281c"  
  }  
]
```



Leveraging the Trace API endpoints to obtain the sliced bread information





Locations endpoint (details about locations)

4. Send the location identifiers found in the events response (Step 2) to the Locations endpoint to obtain the specifics for a location.

/v2/locations

Locations

Obtain the location information for the warehouse and the manufacturing facility:

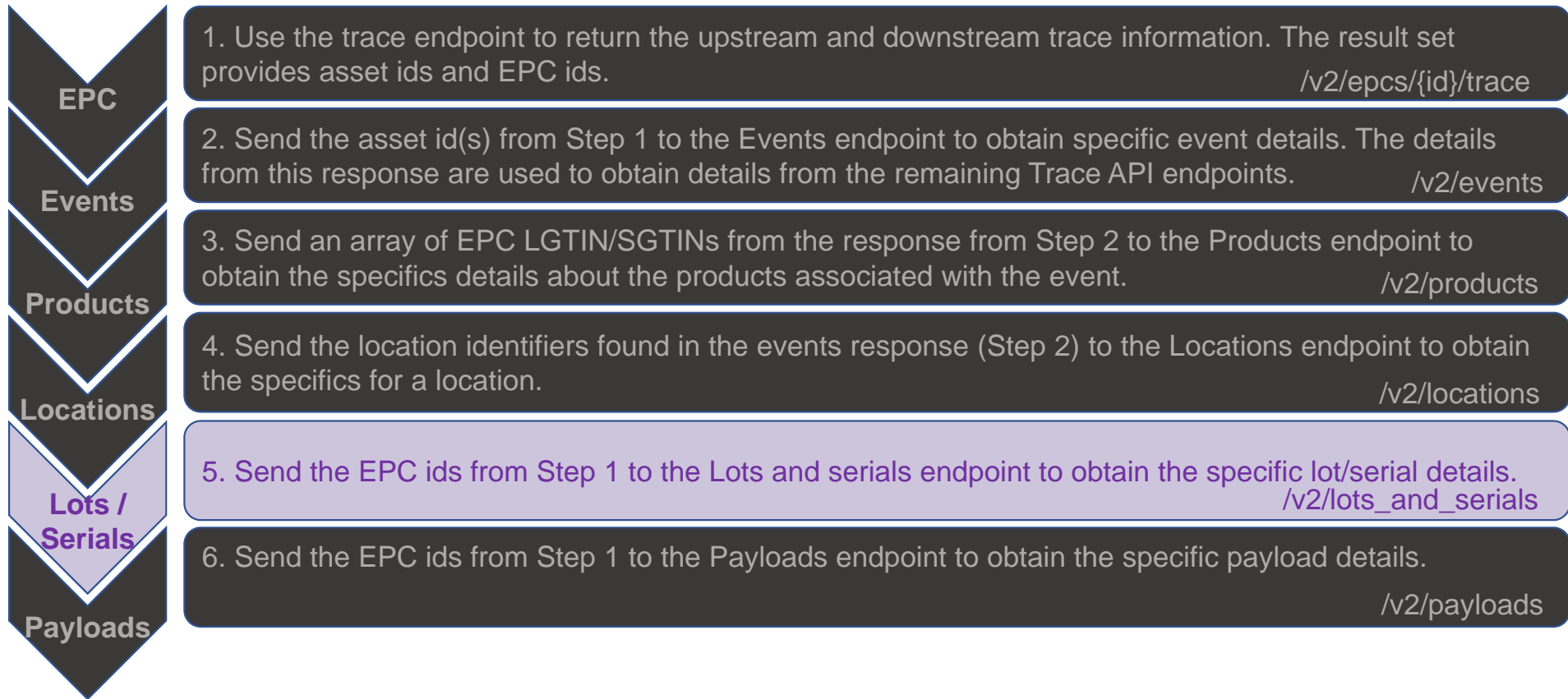
```
curl -X GET
```

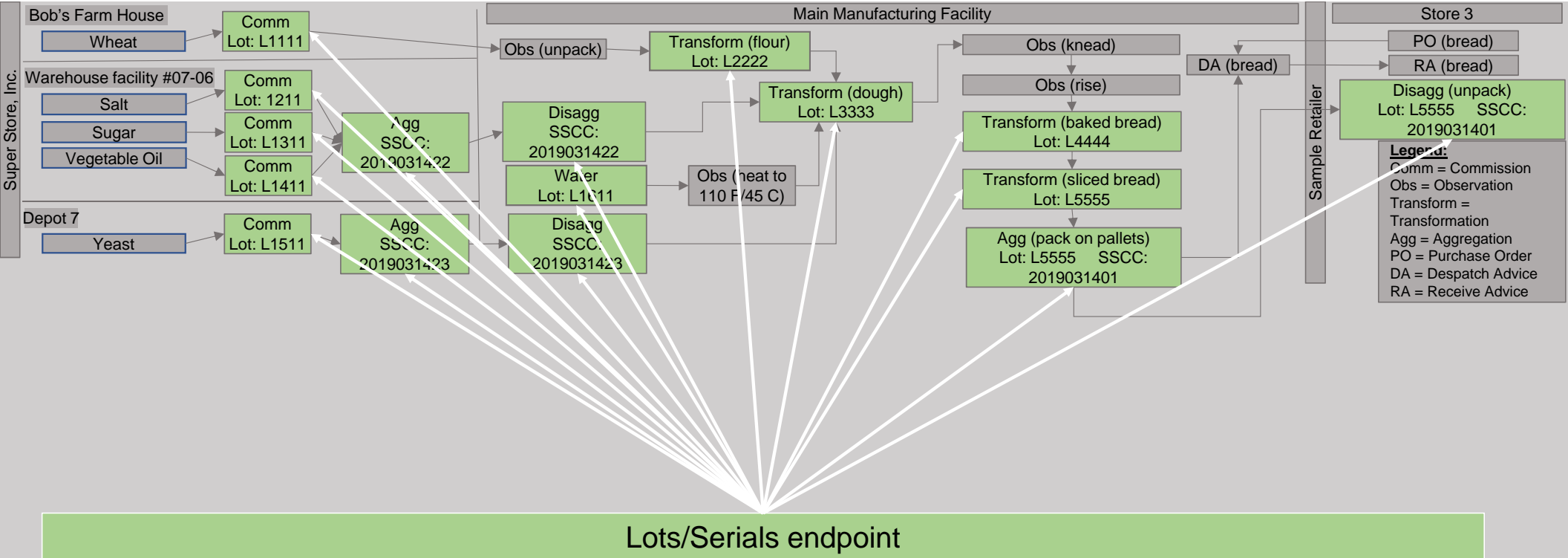
```
'https://food.ibm.com/ift/api/outbound/v2/locations?location_id[]=urn:ibm:ift:location:loc:0002662.DC9574&location_id[]=urn:ibm:ift:location:loc:0548138542.PFGs' -H 'Authorization: Bearer ey...Kw'
```

```
{
  "locations": [
    {
      "id": "urn:ibm:ift:location:loc:0002662.DC9574",
      "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",
      "party_role_code": "WAREHOUSE AND OR DEPOT",
      "party_name": "Warehouse facility #07-06",
      "city": "Fontana",
      "country_code": "US",
      "postal_code": "22335",
      "state": "CA"
    },
    {
      "id": "urn:ibm:ift:location:loc:0548138542.PFGs",
      "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",
      "party_role_code": "MANUFACTURER OF GOODS",
      "party_name": "Main Manufacturing Facility",
      "city": "Dublin",
      "country_code": "US",
      "postal_code": "43016",
      "state": "Ohio"
    }
  ]
}
```



Leveraging the Trace API endpoints to obtain the sliced bread information

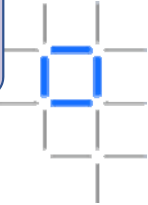






Lots /
Serials

5. Send the EPC ids from Step 1 to the Lots and serials endpoint to obtain the specific lot/serial details.
[/v2/lots_and_serials](#)



Obtain lots and serials information for the EPC ids:

```
curl -X GET
```

```
'https://food.ibm.com/ift/api/outbound/v2/lots_and_serials?lot_or_serial_id[]=urn:ibm:ift:lot:class:0883655000.wthX.L1611&lot_or_serial_id[]=urn:ibm:ift:lot:class:0002662.iohF.L4444&lot_or_serial_id[]=urn:ibm:ift:lot:class:0002662.Ifyl.L5555' -H 'Authorization: Bearer ey...Kw'
```

```
{
  "lots_and_serials":
  [
    { "id": "urn:ibm:ift:product:lot:class:0883655000.wthX.L1611",
      "product_id": "urn:ibm:ift:product:class:0883655000.wthX"
    },
    { "id": "urn:ibm:ift:product:lot:class:0002662.iohF.L4444",
      "product_id": "urn:ibm:ift:product:class:0002662.iohF"
    },
    {
      "id": "urn:ibm:ift:product:lot:class:0002662.Ifyl.L5555",
      "product_id": "urn:ibm:ift:product:class:0002662.Ifyl"
    }
  ]
}
```



Payloads

6. Send the EPC ids from Step 1 to the Payloads endpoint to obtain the specific payload details.
/v2/payloads



Obtain payload information for the EPC ids:

```
curl -X GET
```

```
'https://food.ibm.com/ift/api/outbound/v2/payloads?epc_id[]=urn:ibm:ift:lot:class:0883655000.wthX.L1611&epc_id[]=urn:ibm:ift:lot:class:0002662.iohF.L4444&epc_id[]=urn:ibm:ift:lot:class:0002662.Ifyl.L5555' -H 'Authorization: Bearer ey...Kw'
```

```
{
  "payloads":
  [
    {
      "id": "f38b3c58-aceb-4093-b1ff-8552675fc83a",
      "org_id": "38d1b14e-6af4-4734-ae6e-d788fbf84283",
      "payload_type_uri": "urn:ibm:gbs:gsf:temperature:ambient:digi",
      "payload_timestamp": "2018-07-24T15:38:14.000Z",
      "payload_content_type": "application/json",
      "event_ids": [
        "11111"
      ],
      "epc_ids": [
        "urn:ibm:ift:lot:class:0883655000.wthX.L1611",
        "urn:ibm:ift:lot:class:0002662.Ifyl.L5555"
      ],
      "location_ids": [],
      "sub_location_ids": [],
      "payload": "{\"payload\":{\"sensor_id\":\"44644\",\"temp_data\": [{\"temp\":39,\"uom\":\"FAH\",\"timestamp\":\"20180724T13:54:43\"}, {\"temp\":39,\"uom\":\"FAH\",\"timestamp\":\"2018-07-24T14:03:24\"}],}}}"
    }
  ]
}
```