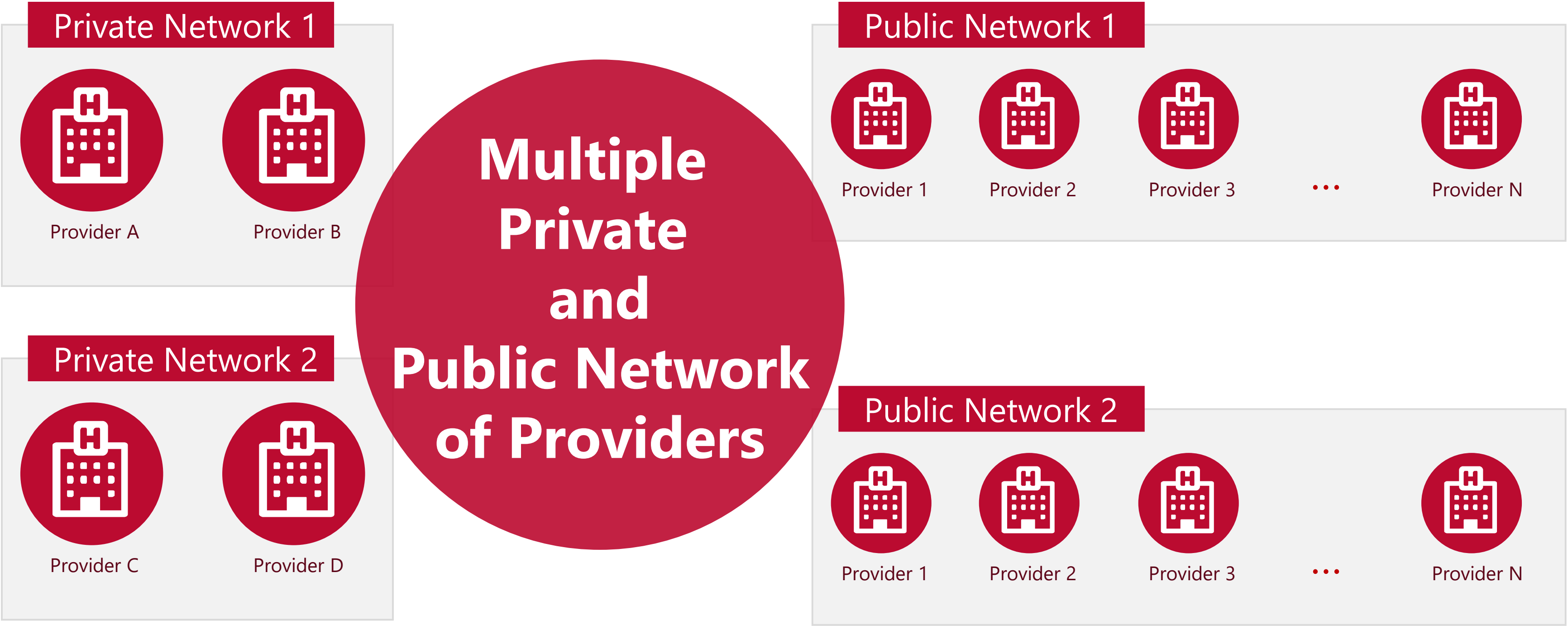
RosterNet Architecture

Federation



Private Network 1

Provider A

Provider B

Private Network 2

Provider C

Provider D

Multiple Private and Public Network of Providers

Public Network 1

Provider 1

Provider 2

Provider 3

…

Provider N

Public Network 2

Provider 1

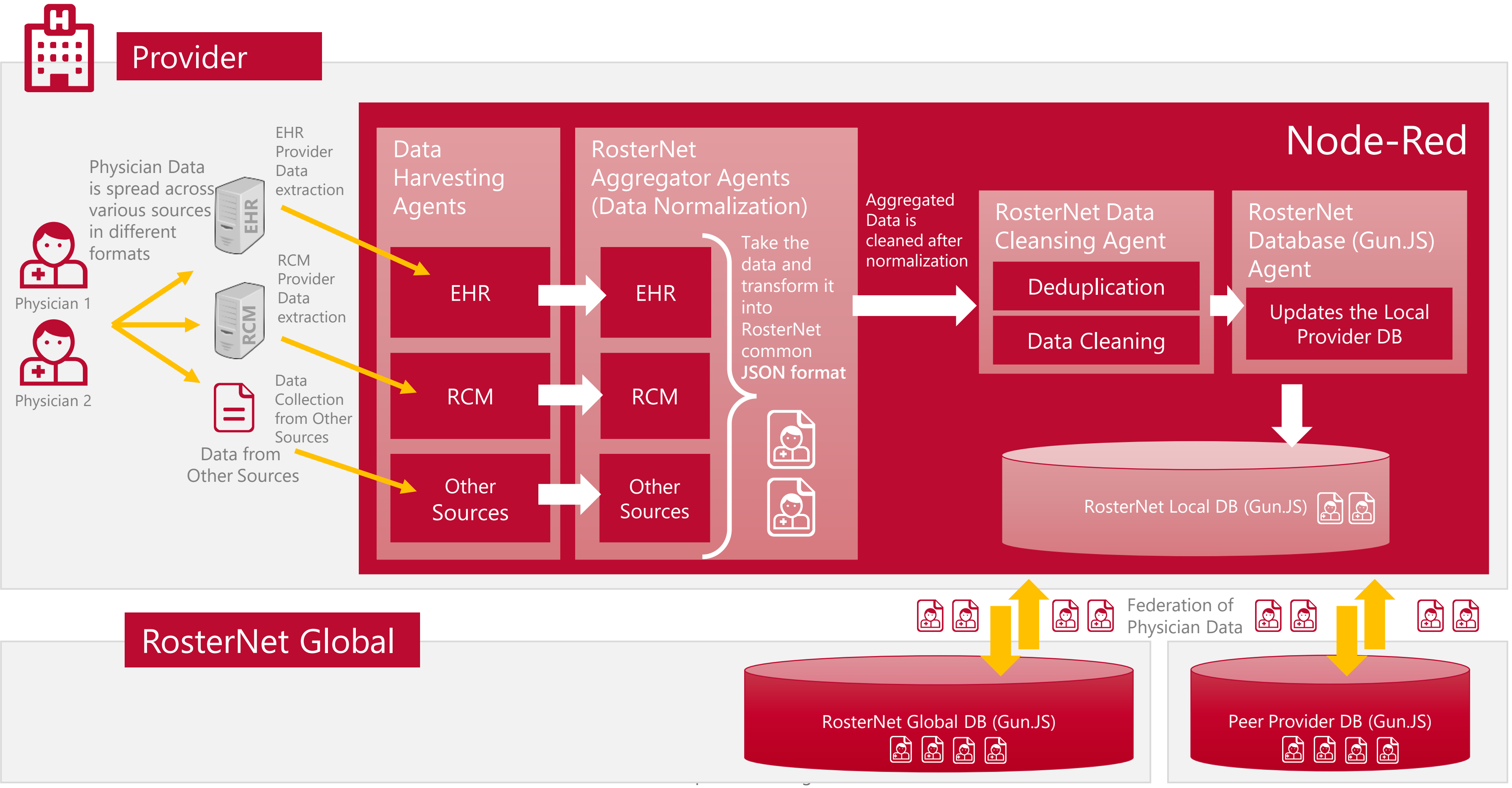
Provider 2

Provider 3

…

Provider N

Roster Data Producer



Provider

Physician Data is spread across various sources in different formats

Physician 1

Physician 2

EHR Provider Data extraction

RCM Provider Data extraction

Data Collection from Other Sources

Data from Other Sources

Node-Red

Data Harvesting Agents

- EHR
- RCM
- Other Sources

RosterNet Aggregator Agents (Data Normalization)

- EHR
- RCM
- Other Sources

Take the data and transform it into RosterNet common **JSON format**

Aggregated Data is cleaned after normalization

RosterNet Data Cleansing Agent

- Deduplication
- Data Cleaning

RosterNet Database (Gun.JS) Agent

- Updates the Local Provider DB

RosterNet Local DB (Gun.JS)

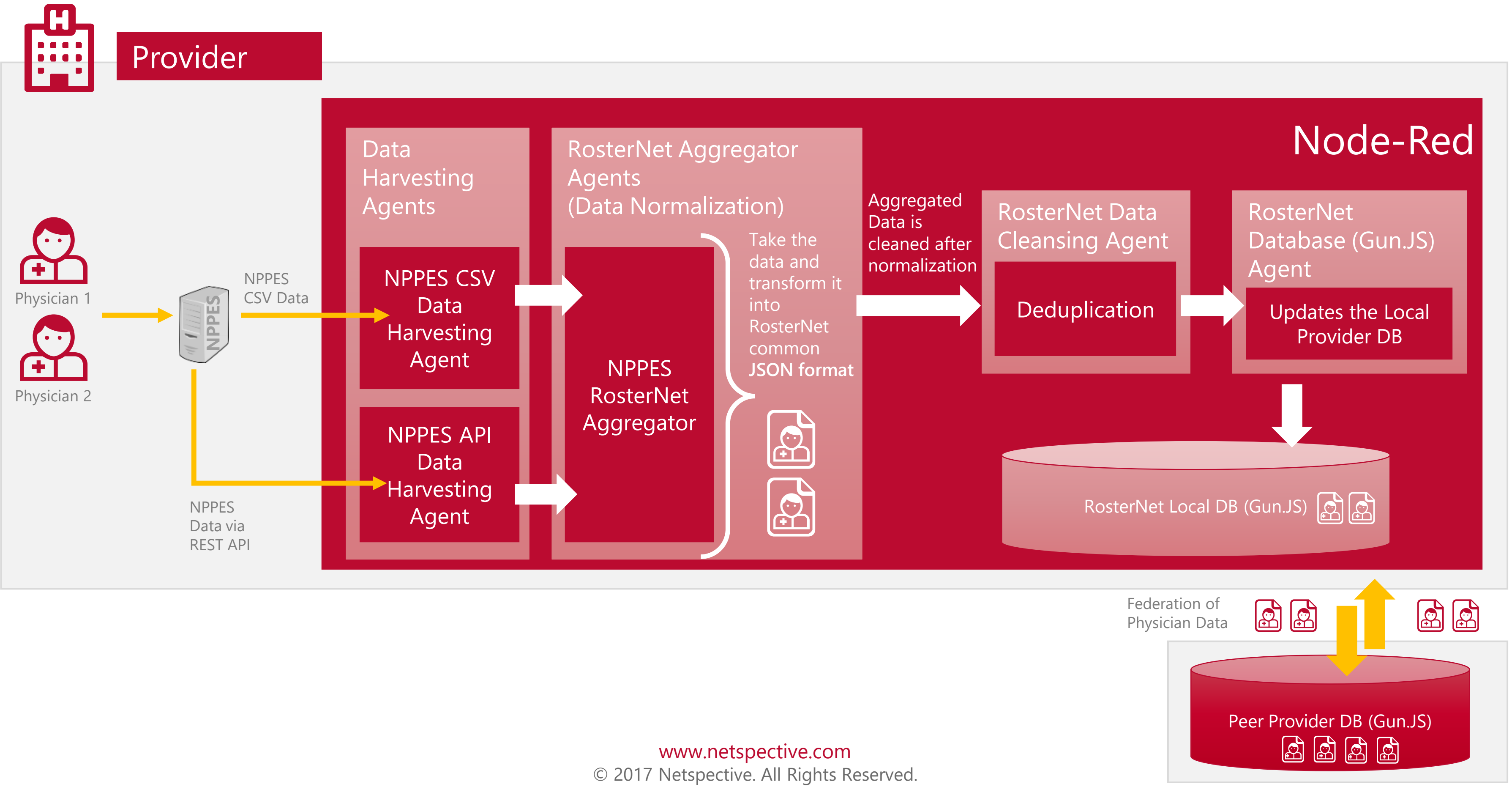
Federation of Physician Data

RosterNet Global

RosterNet Global DB (Gun.JS)

Peer Provider DB (Gun.JS)

Roster Data Producer - Example



Provider

Node-Red

Data Harvesting Agents

NPPES CSV Data Harvesting Agent

NPPES API Data Harvesting Agent

RosterNet Aggregator Agents (Data Normalization)

NPPES RosterNet Aggregator

Take the data and transform it into RosterNet common **JSON format**

Aggregated Data is cleaned after normalization

RosterNet Data Cleansing Agent

Deduplication

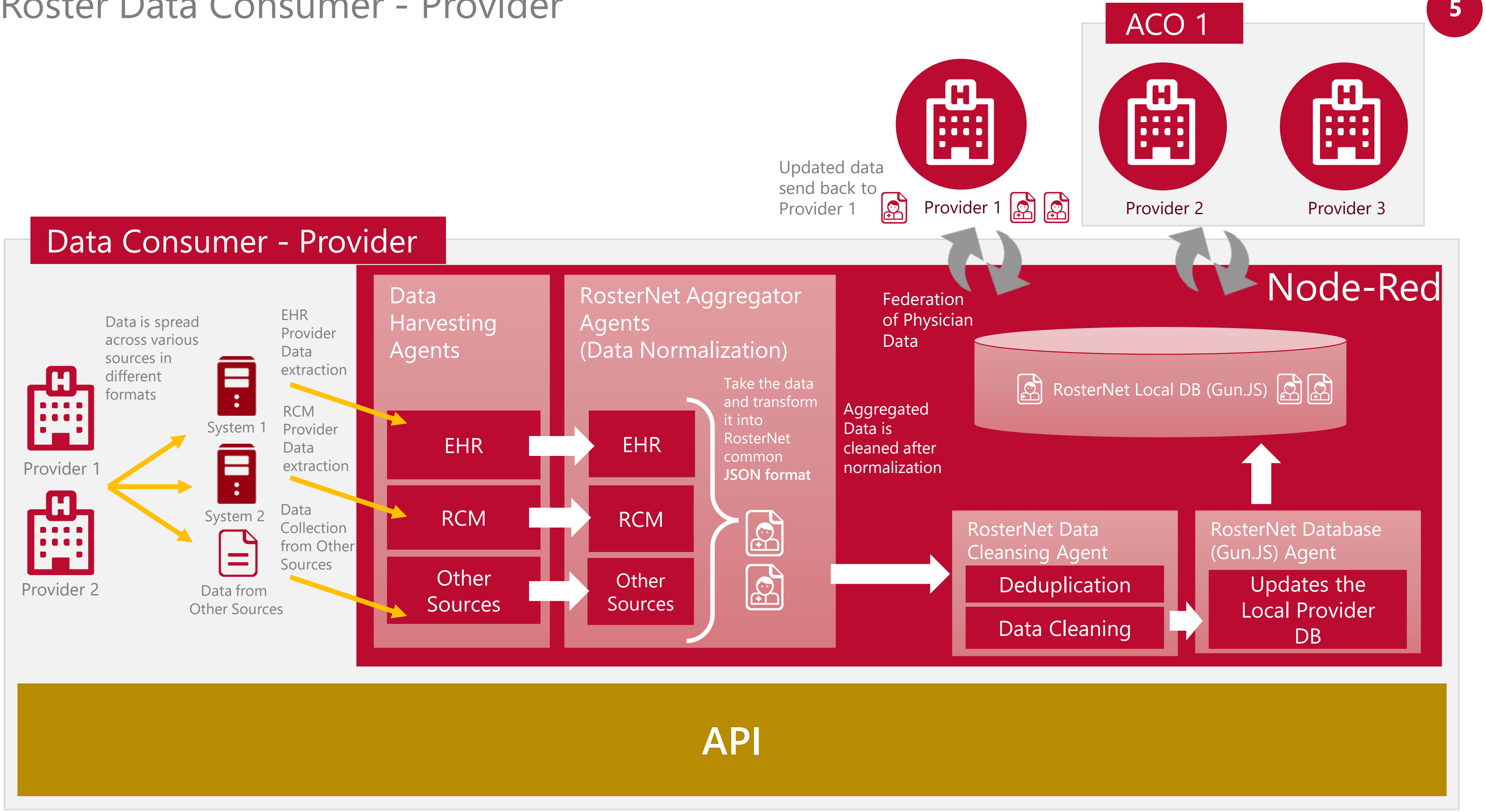
RosterNet Database (Gun.JS) Agent

Updates the Local Provider DB

Physician 1
Physician 2
NPPES
NPPES CSV Data
NPPES Data via REST API
RosterNet Local DB (Gun.JS)
Federation of Physician Data
Peer Provider DB (Gun.JS)

www.netspective.com

Roster Data Consumer - Provider



ACO 1

Provider 1

Provider 2

Provider 3

Updated data send back to Provider 1

Data Consumer - Provider

Node-Red

Provider 1

Provider 2

Data is spread across various sources in different formats

System 1

System 2

Data from Other Sources

EHR Provider Data extraction

RCM Provider Data extraction

Data Collection from Other Sources

Data Harvesting Agents

- EHR
- RCM
- Other Sources

RosterNet Aggregator Agents (Data Normalization)

- EHR
- RCM
- Other Sources

Take the data and transform it into RosterNet common **JSON format**

Federation of Physician Data

RosterNet Local DB (Gun.JS)

Aggregated Data is cleaned after normalization

RosterNet Data Cleansing Agent

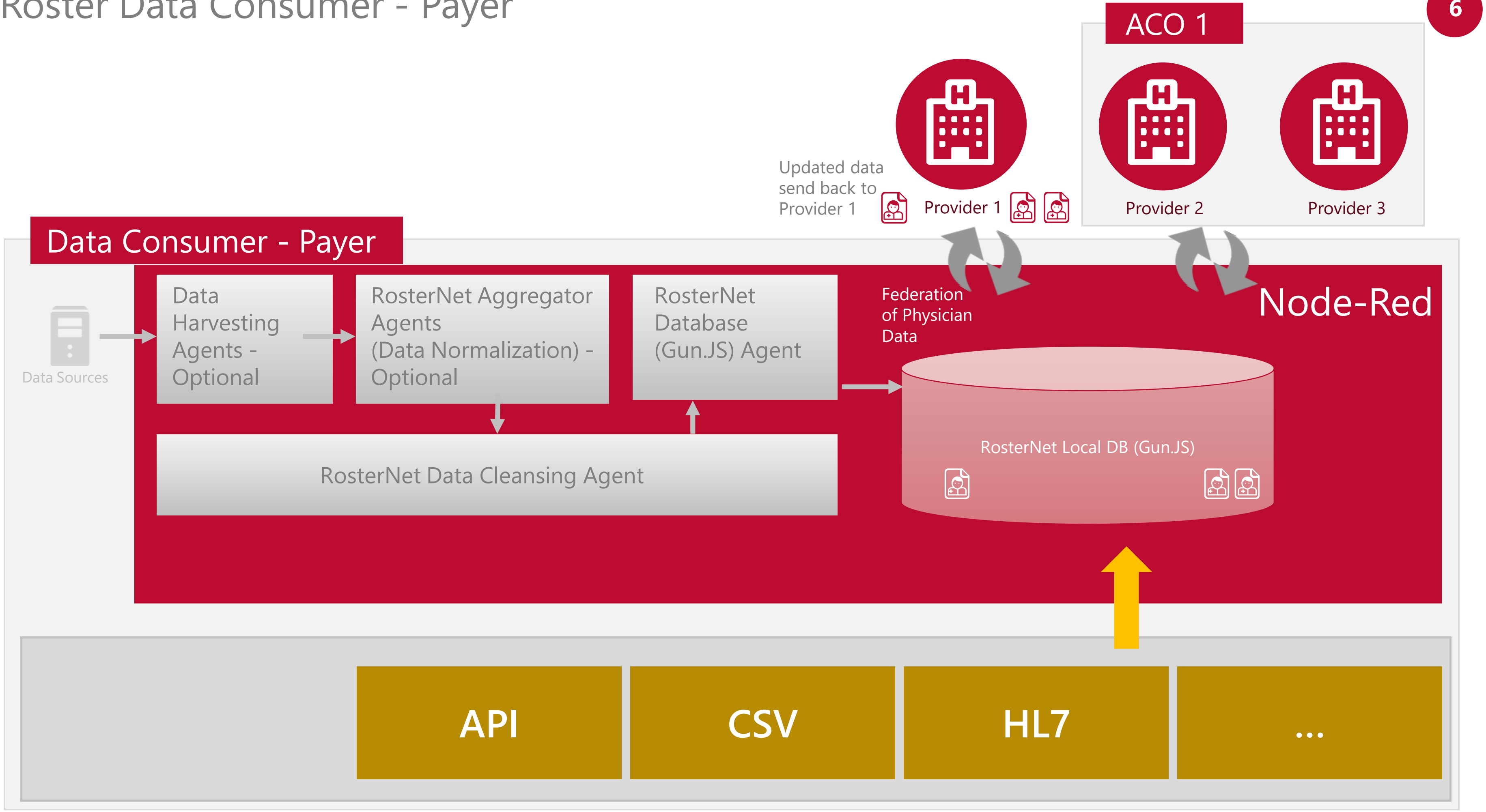
- Deduplication
- Data Cleaning

RosterNet Database (Gun.JS) Agent

- Updates the Local Provider DB

API

Roster Data Consumer - Payer



ACO 1

Updated data send back to Provider 1

Provider 1

Provider 2

Provider 3

Data Consumer - Payer

Node-Red

Data Sources

Data Harvesting Agents - Optional

RosterNet Aggregator Agents (Data Normalization) - Optional

RosterNet Database (Gun.JS) Agent

Federation of Physician Data

RosterNet Local DB (Gun.JS)

RosterNet Data Cleansing Agent

API

CSV

HL7

...

Data Source

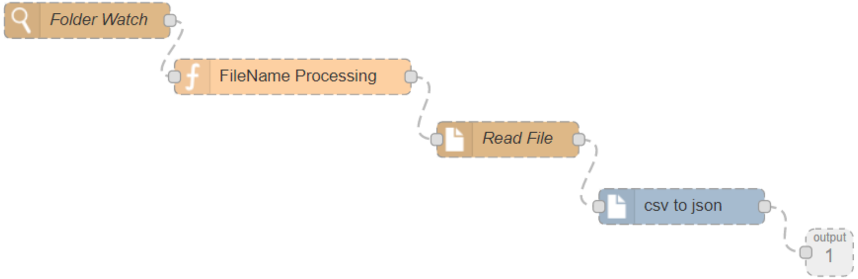
- Phase 1 :
  - 1 Provider Data Source
  - NPPES CSV Data(Available for Download) and Data from NPPES API calls
- Phase 2 and future Phases: Additional Provider or data source integration

Environment

- The RosterNet Provider environment is based on Node-Red (Node JS based)
- Pre-requisites
  - Node JS (https://nodejs.org/en/, https://github.com/nodejs/node)
  - Gun JS (http://gun.js.org/, https://github.com/amark/gun)
  - Node-Red (https://nodered.org/, https://github.com/node-red/node-red)
- RosterNet Agents are developed as flows and subflows in Node-Red.
- For creating flows and subflows, new nodes were developed for functionalities like JSON Transformation, Filtering, Deduplication, Gun.JS Client etc. and used along with existing Node-Red nodes.
- Github

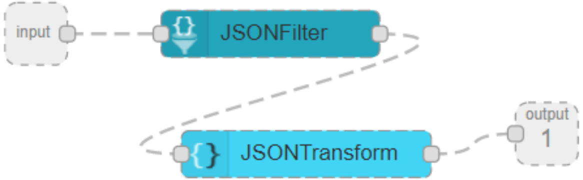
Data Harvesting Agents

- Specific to each Data Source, there should be a Data Harvesting Agent which pulls data from the source and provide to the Respective RosterNet Aggregator Agents.
- For Example, the **CSV Data Harvesting Agent** flow below watches a folder for CSV files, Read the file and provide the data as JSON.
- Based on the Data and Data Source, we will need to create as many Data Harvesting Agents as possible.



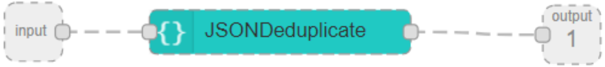
RosterNet Aggregator Agents

- RosterNet Aggregator Agents transforms incoming data in custom format to RosterNet Common JSON format.
- Specific to each Data Source, we should have a RosterNet Aggregator Agent to transform the raw data that comes in to RosterNet Common JSON format.
- For Example, the **RosterNet Aggregator** flow below Filters out unwanted fields from the input and transforms the data to RosterNet Common JSON format.



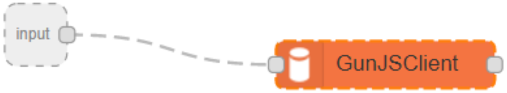
RosterNet Cleansing Agent

- RosterNet Cleansing Agents performs the data cleaning operations on the data in RosterNet Common JSON format.
- This is common for all data sources.
- For Example, the **RosterNet Data Cleansing Agent** flow below performs deduplication on Data in RosterNet Common JSON format.

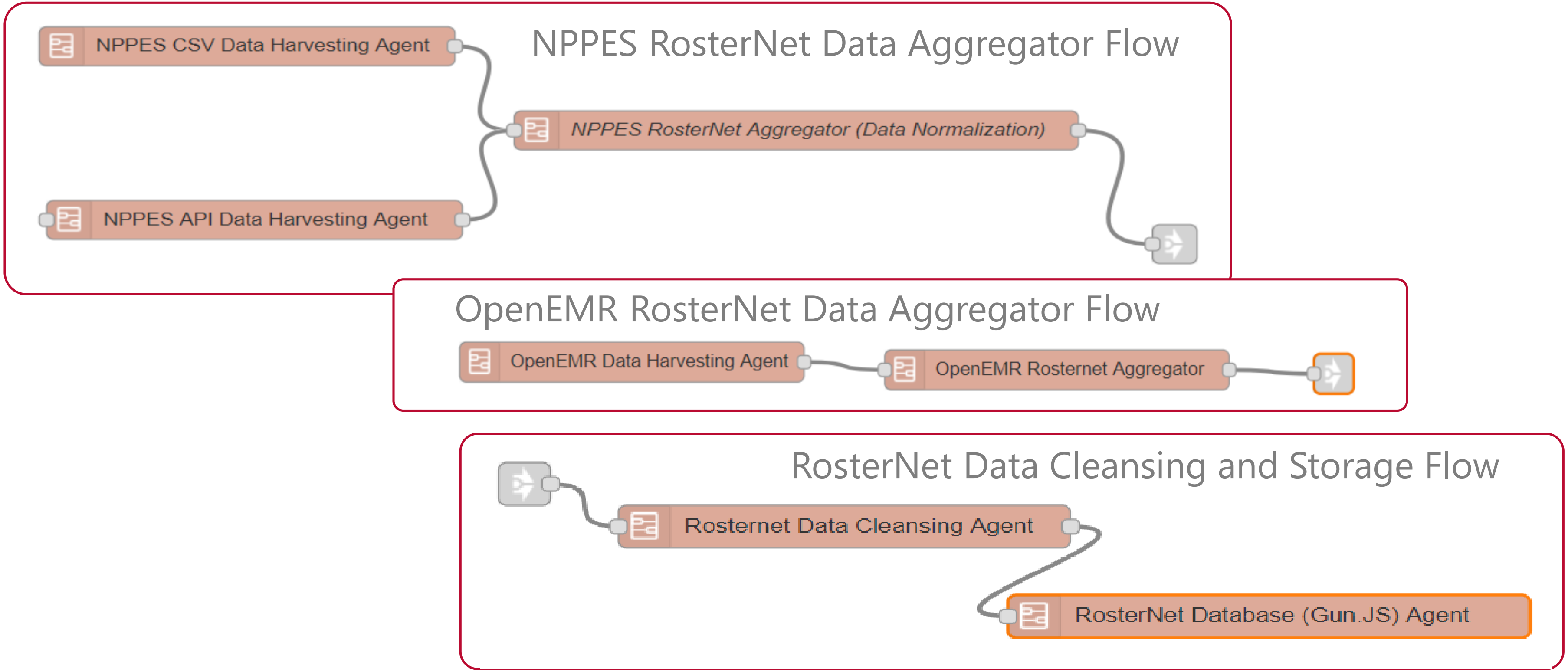


RosterNet Database (Gun.JS) Agent

- RosterNet Database Agent pushes to the local provider DB in RosterNet Common JSON format from where it gets federated.
- This is common for all data sources.
- For Example, the **RosterNet Database (Gun.JS) Agent** flow below stores data to the local Gun.JS server as well as federates the data across its peers.



Overall Flow - Example



NPPES RosterNet Data Aggregator Flow

NPPES CSV Data Harvesting Agent

NPPES API Data Harvesting Agent

NPPES RosterNet Aggregator (Data Normalization)

OpenEMR RosterNet Data Aggregator Flow

OpenEMR Data Harvesting Agent

OpenEMR Rosternet Aggregator

RosterNet Data Cleansing and Storage Flow

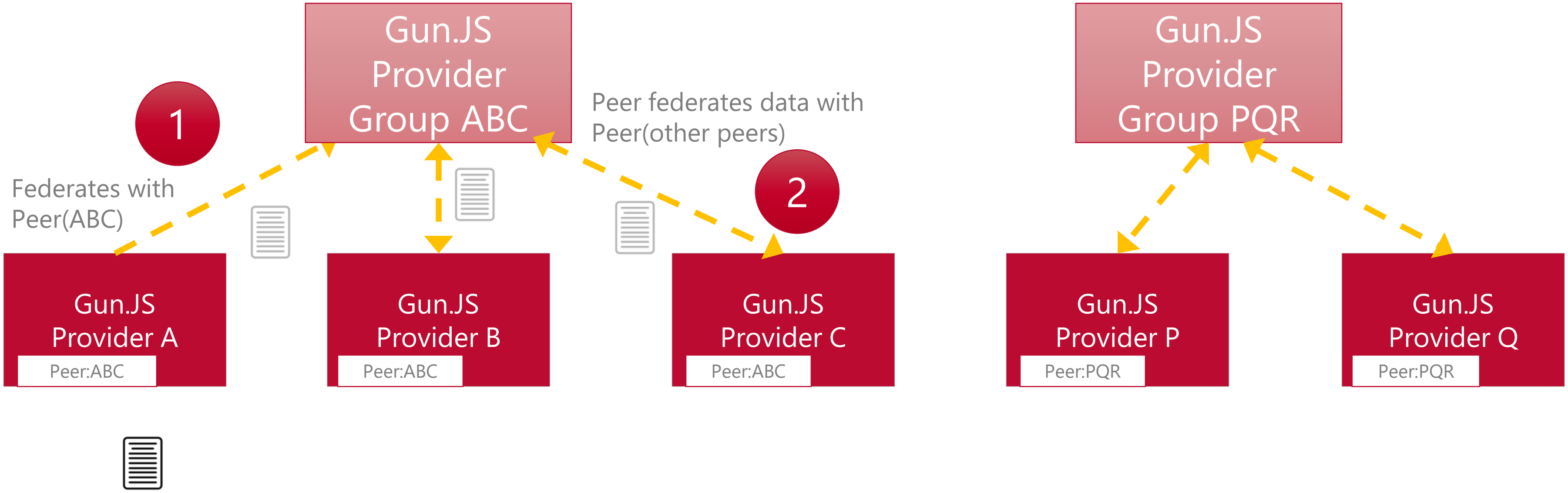
Rosternet Data Cleansing Agent

RosterNet Database (Gun.JS) Agent

APPENDIX

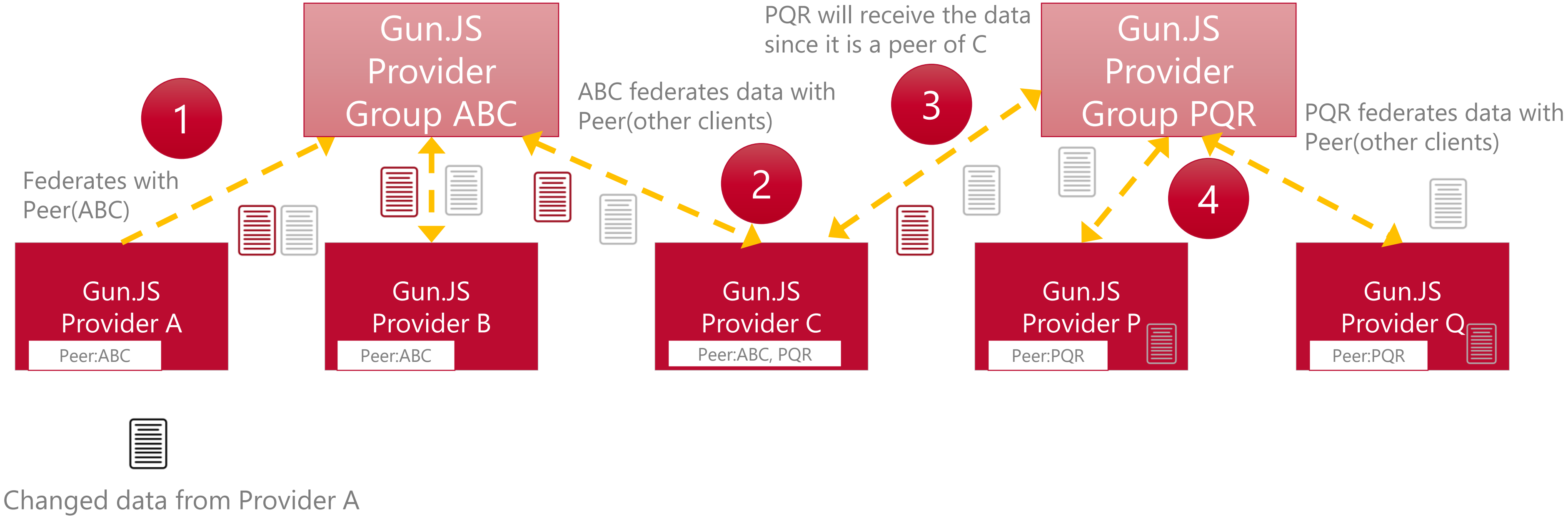
Gun.JS Data Federation

PQR, P and Q will not receive the data since it is not part of the peer network of ABC



Changed data from Provider A

Gun.JS Data Federation


Gun.JS Provider Group ABC
PQR will receive the data since it is a peer of C
Gun.JS Provider Group PQR
1
ABC federates data with Peer(other clients)
3
PQR federates data with Peer(other clients)
Federates with Peer(ABC)
2
4
Gun.JS Provider A
Peer:ABC
Gun.JS Provider B
Peer:ABC
Gun.JS Provider C
Peer:ABC, PQR
Gun.JS Provider P
Peer:PQR
Gun.JS Provider Q
Peer:PQR
Changed data from Provider A