

Single Day Event

Optimized Operational Data – Simplifying with Higher Speed and Productivity

Manish Pathak,
Novartis, Hyderabad



Disclaimer

The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of Novartis.

Novartis does not guarantee the accuracy or reliability of the information provided herein.



Contents

- ❑ Power of Visualizations
- ❑ Data Digitalization Journey
- ❑ Data Capture Technology Stack
- ❑ Visualization Framework
- ❑ Visualization Library: Setup
- ❑ Visualization Library: Setup & Example
- ❑ Visualization Library: Example code
- ❑ Exploring Visualizations
- ❑ New-Gen Dashboard challenges
- ❑ Visualization – Use cases

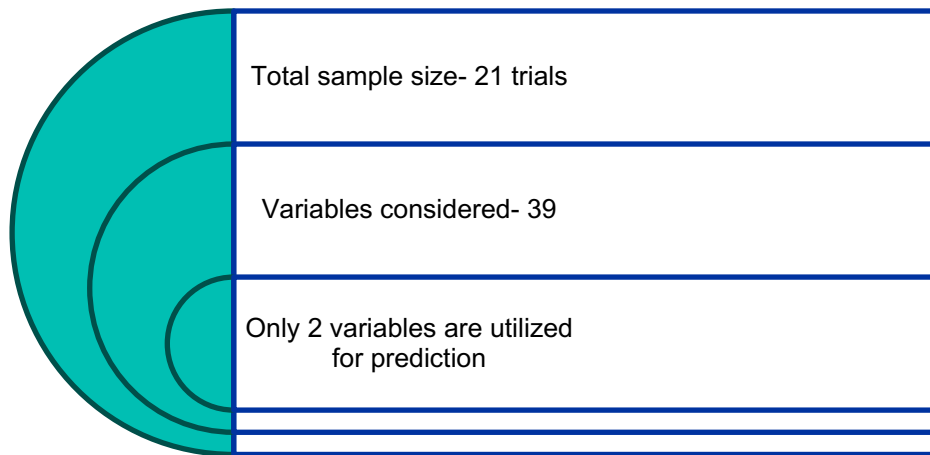


Power of Visualizations

Harnessing technology to facilitate and enhance clinical trial teams experience and active workflows

- Integrating data science with visualization to better serve our pharma community and colleagues.
- Using insightful visualizations to transform storytelling and optimize problem-solving.
- Transitioning from reaction to action by harnessing computation power to minimize noise through Predictive Analytics Engine.

Example: Simple visualization with Waterfall Chart for DBL predictions. Flagging lagging trials and CTT attention will help to achieve desired KPI's.

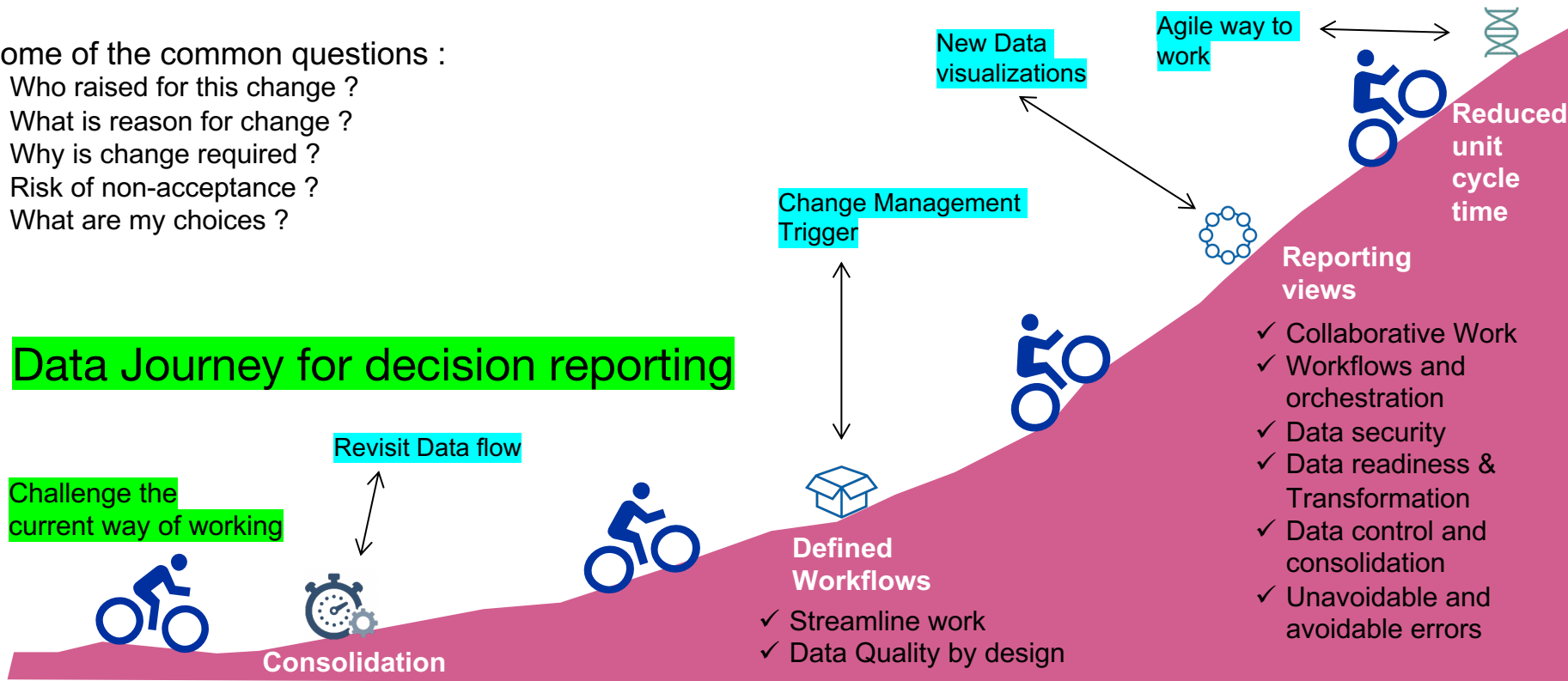




Some of the common questions :

- Who raised for this change ?
- What is reason for change ?
- Why is change required ?
- Risk of non-acceptance ?
- What are my choices ?

Data Journey for decision reporting





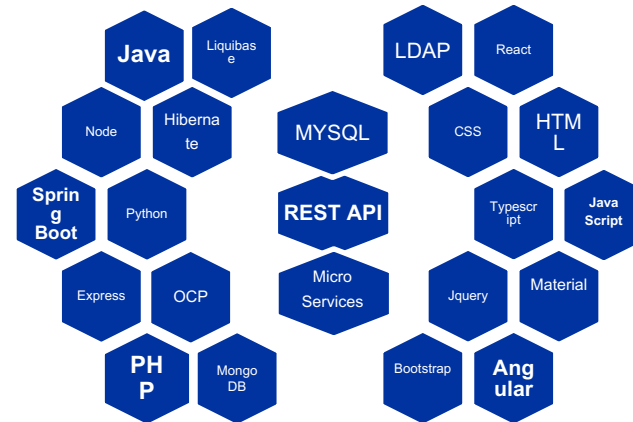
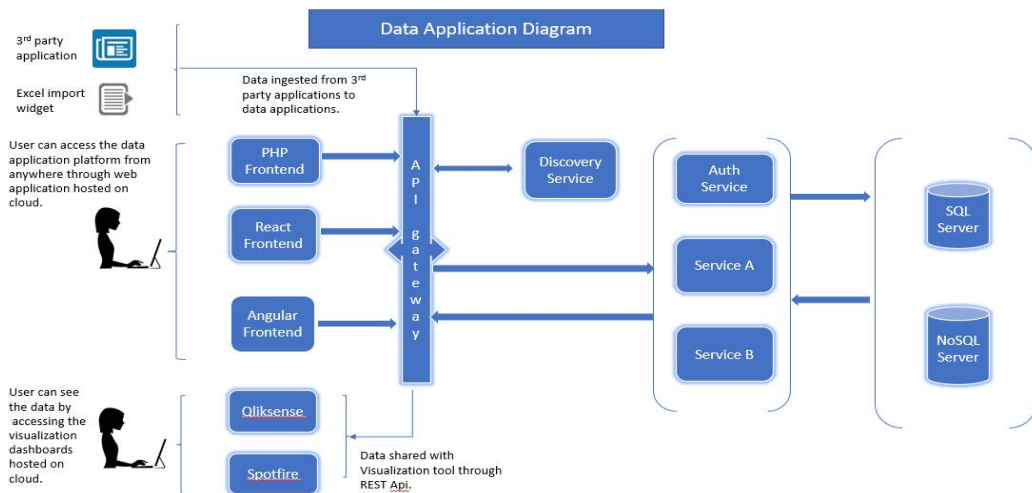
Data Capture Technology Stack



Clinical Data Operational data is captured using below technology stack. These technologies help to create a user to input operational data to a database solution

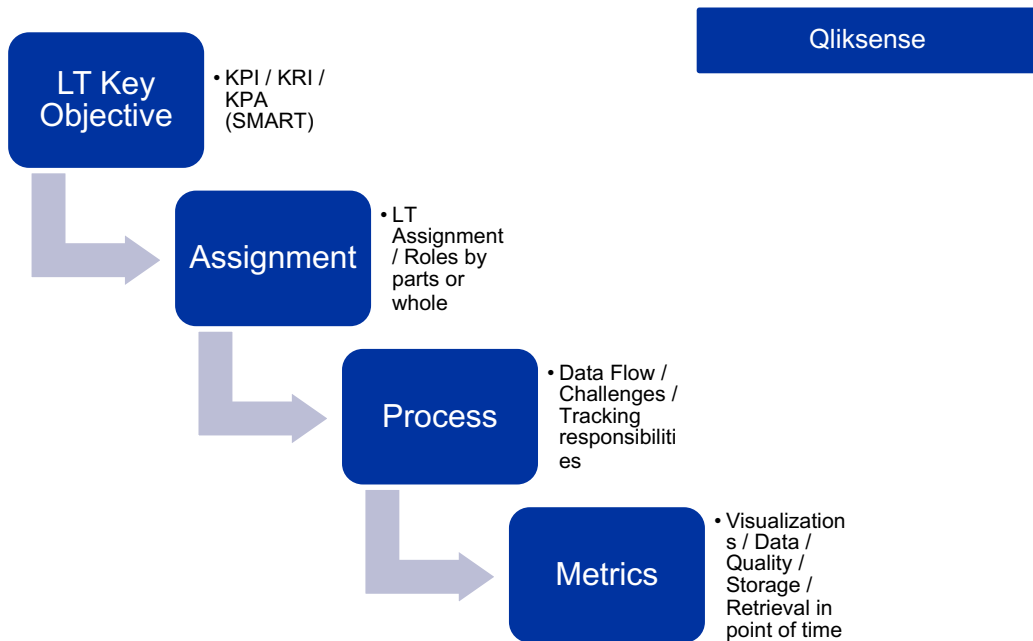
Architecture diagram for applications with centralized sign-on and workflow orchestration supported.

Data using API's are exchanged with other data bases and visualization platforms





Each Visualization representing the metrics is linked to the governance framework. Post which Qlik sense / Spotfire are primarily used for Visualizations within Operational space for below reasons



<u>Definition:</u> Metrics Definition with	<u>Thresholds:</u> Green: $\geq 80\%$ Yellow (<10%): $> 70\% \ \& \ < 80\%$ Red ($\geq 10\%$): $\leq 70\%$
<u>I/E Criteria:</u> Criteria for I/E Listed. Reasoning if required	<u>High Level Commentary:</u> Metric performance /Tracking regular intervals / comments / LT asks



<https://github.com/apache/echarts>

JS setup.js

```
#Get Apache ECharts
#Download from the official website
npm install echarts --save
#CDN: jsDelivr CDN

# Build
# Install the dependencies from NPM:
npm install

# Rebuild source code immediately in watch mode when changing the source code.
# It opens the `./test` directory, and you may open `-cases.html` to get the list
# of all test cases.
# If you wish to create a test case, run `npm run mktest:help` to learn more.
npm run dev

# Check the correctness of TypeScript code.
npm run checktype

# If intending to build and get all types of the "production" files:
npm run release
```



Visualization Library: Setup & Example



Installing from npm

If you're developing a web application using Node, you can install D3 via yarn, npm, pnpm, or your preferred package manager.

```
yarn  npm  pnpm
```

```
pnpm add d3
```

```
bash
```

You can then load D3 into your app as:

```
import * as d3 from "d3";
```

```
js
```

You can instead import specific symbols if you prefer:

```
import {select, selectAll} from "d3";
```

```
js
```

Alternatively you can install and import from D3 submodules:

```
import {mean, median} from "d3-array";
```

```
js
```

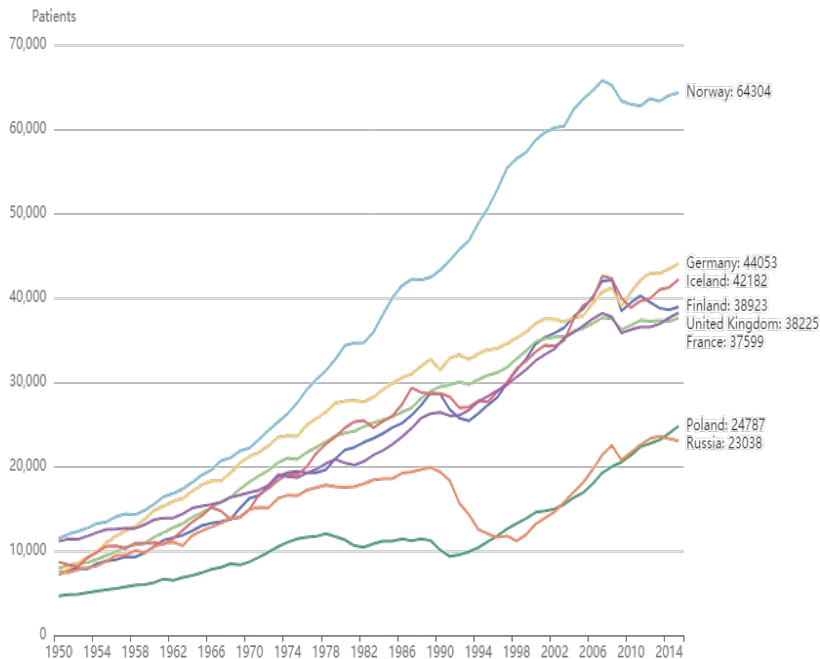
TypeScript declarations are available via DefinitelyTyped.

```
d3.jsx
```

```
import * as d3 from "d3";
```

```
export default function LinePlot({
  data,
  width = 640,
  height = 400,
  marginTop = 20,
  marginRight = 20,
  marginBottom = 20,
  marginLeft = 20
}) {
  const x = d3.scaleLinear([0, data.length - 1],
    [marginLeft, width - marginRight]);
  const y = d3.scaleLinear(d3.extent(data),
    [height - marginBottom, marginTop]);
  const line = d3.line((d, i) => x(i), y);
  return (
    <svg width={width} height={height}>
      <path fill="none" stroke="currentColor"
        stroke-width="1.5" d={line(data)} />
      <g fill="white" stroke="currentColor" stroke-width="1.5">
        {data.map((d, i)
          => (<circle key={i} cx={x(i)} cy={y(d)} r="2.5" />))}
      </g>
    </svg>
  );
}
```

No of patients enrolled in total by countries



JS eCharts.js

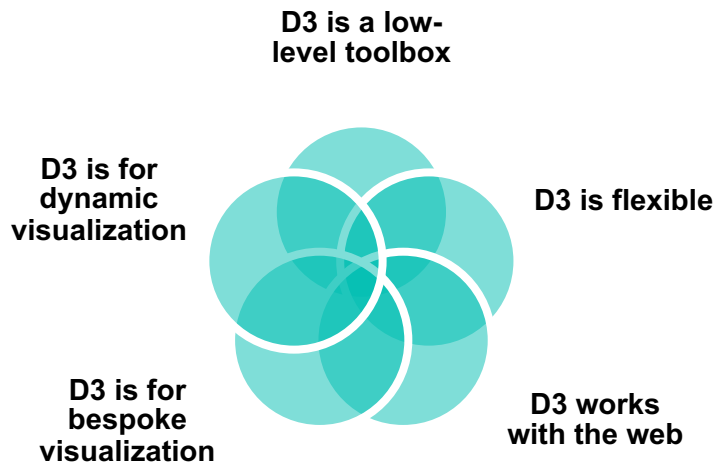
```
function run(_rowData: any) {
  const countries = [
    'Finland', 'France', 'Germany', 'Iceland', 'Norway',
    'Poland', 'Russia', 'United Kingdom'
  ];
  const datasetWithFilters: echarts.DatasetComponentOption[] = [];
  const seriesList: echarts.SeriesOption[] = [];
  echarts.util.each(countries, function (country) {
    var datasetId = 'dataset_' + country;
    datasetWithFilters.push({
      id: datasetId,
      fromDatasetId: 'dataset_raw',
      transform: {
        type: 'filter',
        config: {
          and: [
            { dimension: 'Year', gte: 1950 }, { dimension: 'Country', '=': country }
          ]
        }
      }
    });
    seriesList.push({
      type: 'line', datasetId: datasetId,
      showSymbol: false, name: country,
      endLabel: {
        show: true,
        formatter: function (params: any) {
          return params.value[3] + ' : ' + params.value[0];
        }
      },
      labelLayout: { moveOverlap: 'shiftY' },
      emphasis: { focus: 'series' },
      encode: {
        x: 'Year', y: 'Patients',
        label: ['Country', 'Patients'], itemName: 'Year', tooltip: ['Patients']
      }
    });
  });
  myChart.setOption(option);
}
```

Reactjs and Angular Framework will help in exploring next-gen visualization. Apache ECharts is a free and powerful charting.

1. Versatility
2. Interactive Visualizations
3. Customizability
4. Compatibility
5. Rich Animation
6. Big Data Support
7. Powerful Rendering Engine
8. Professional Data Analysis



D3.js: D3 (or D3.js) is a free, open-source JavaScript library for visualizing data. Its low-level approach built on web standards offers unparalleled flexibility in authoring dynamic, data-driven graphics



Some of the challenges for New Gen Dashboards

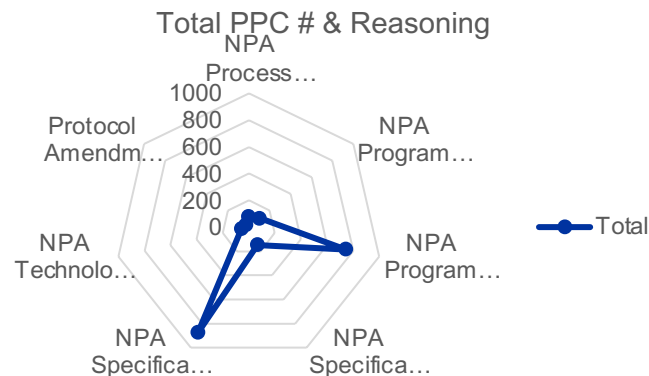
1. Global view
2. Navigation
3. Business objectives / KPI / KPA / KRI
4. Decreasing order of relevancy
5. Near real time
6. Triggers / Reminders
7. Prediction analytics
8. Handle Objectivity





- Post Database delivery Clinical teams request for changed. These changes are clubbed under Post Production Changes (PPC). PPC changes leads to delays and increase cost with efforts. Visualizations to PPC and its associated CR's may help for collective reasoning and collective workgroups can be formed.
 - Database approach to PPC logging using workflow app for quality data capture and retrieval over time.
 - Text Analytics on comments / reasoning using AI.
 - Group terminologies and reasons under defined category.
 - Plot the category with counts over time. Use Pareto to handle asserting relationship.
 - Workstreams use this data for PPC process revision and deep diving

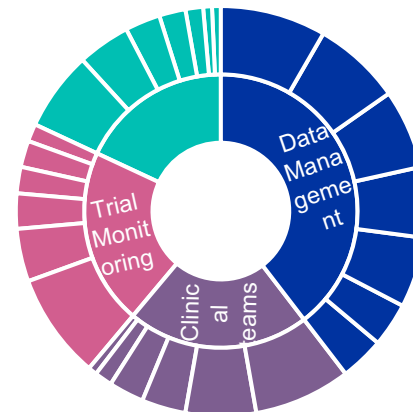
- Trial milestones can be predicted using AI on current and historic data. This visualized will help CTT for immediate action and tracking for milestones lag.



- Post Database Lock for unseen situations & critical cases Clinical teams takes decision to Unlock the database called Database Unlocks (DBUL). This leads to delays in DBL, work impact to multiple teams and finally increased cost with efforts. Visualizations to DBUL help in future avoidance and handling similar cases before DBL's.

- Database approach to DBUL logging using workflow app for quality data capture and data retrieval over time.
- Text Analytics on comments / reasoning using AI.
- Group terminologies and reasons under defined category.
- Plot the category with counts over time and associated teams for avoidable cases.
- Workstreams use this data for DBUL process revision and deep diving

DBUL # & reasons w/t associated teams

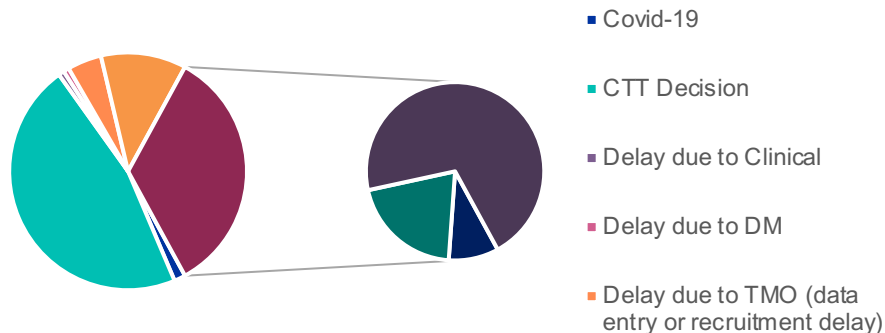


- Portfolio submission delays can be predicted based on AI and then visualized. The Portfolio delay will help Global LT for action



- **Database delays:** CTT works tirelessly to avoid milestone delays esp. Database Locks. Even after all the data tracking, planning and past experiences our teams faces delays. Its important to log these delays for operational efficiencies.
 - Database approach to log Interim and Final DBL milestones dates and reasoning.
 - Delays by sub teams and reasoning helps to understand DBL cases
 - Plot the category with counts over time.
 - End to end Workflow and handshakes to be reviewed.
 - Teams try to bring such events in future planning and reference
 - Metrics process helps esp. in complex and priority trials
- Clinical documents missing can be visualized using meta data for audit / inspection and higher compliances

DBL Delays and Reasoning





References

- To make a stacked area chart: <https://observablehq.com/@d3/stacked-area-chart/2>
- To make a CSV parser to load data: <https://d3js.org/d3-dsv>
- For horizontal position (x): <https://d3js.org/d3-dsv>
- For vertical position (y): <https://d3js.org/d3-scale/linear>
- For ordinal scale and categorical scheme for color: <https://d3js.org/d3-scale/ordinal>
- Stack layout for arranging values: <https://d3js.org/d3-shape/stack>
- For documenting the position encodings: <https://d3js.org/d3-axis>
- For creating SVG elements: <https://d3js.org/d3-selection>



The Global Healthcare Data Science Community

Contact Channels

- 📞 UK +44 1843 609600
- ✉ office@phuse.global
- 🌐 phuse.global

Social Media

- 🐦 [@phusetwitta](https://twitter.com/phusetwitta)
- 📘 [/phusebook](https://facebook.com/phusebook)
- 📺 [/phusetube](https://youtube.com/phusetube)
- 🌐 [/company/phuse](https://linkedin.com/company/phuse)