How can Stata Enable Federated Computing for Decentralized Data Analysis?

Narasimha Raghavan, Paul Lambert, Bjarte Aagnes and Jan F Nygård

Cancer Registry of Norway,
Norwegian Institute of Public Health

10 September 2024

2024 Northern European Stata Conference

Today

- Intro to federated computing
- Share our experiences with federated computing.
- Some preliminary thoughts on integrating federated computing with Stata

Not Today

- Provide a complete solution to integrating Stata with federated computing
- Discuss business or commercial aspects
- Offer in depth technical tutorials

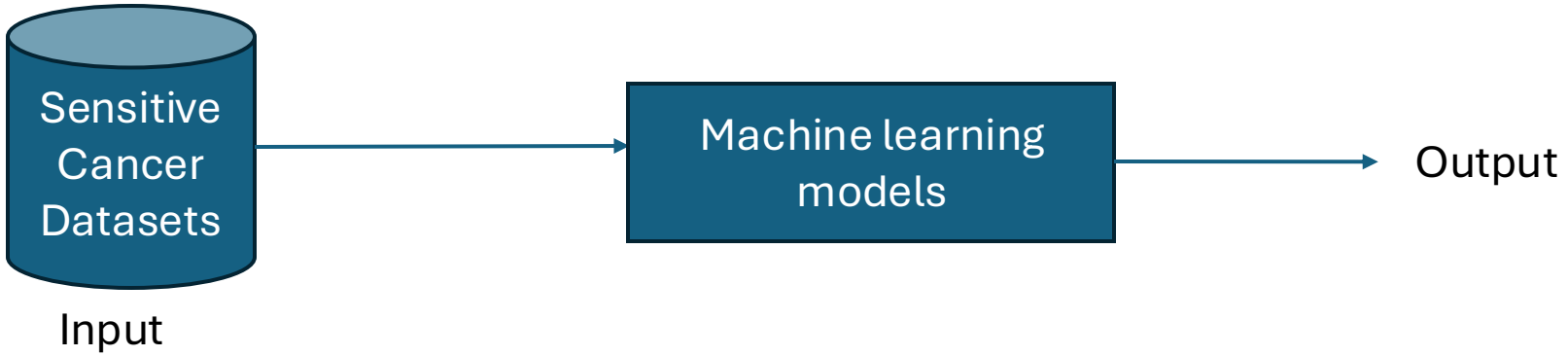
Outline

- Motivation
- Introduction to Federated Computing
- A Federated Computing Example
- Federated Computing Integration with Stata
- Practical limitations
- Demo

Motivation

Why should we care about federated computing?

Data Scarcity Challenge



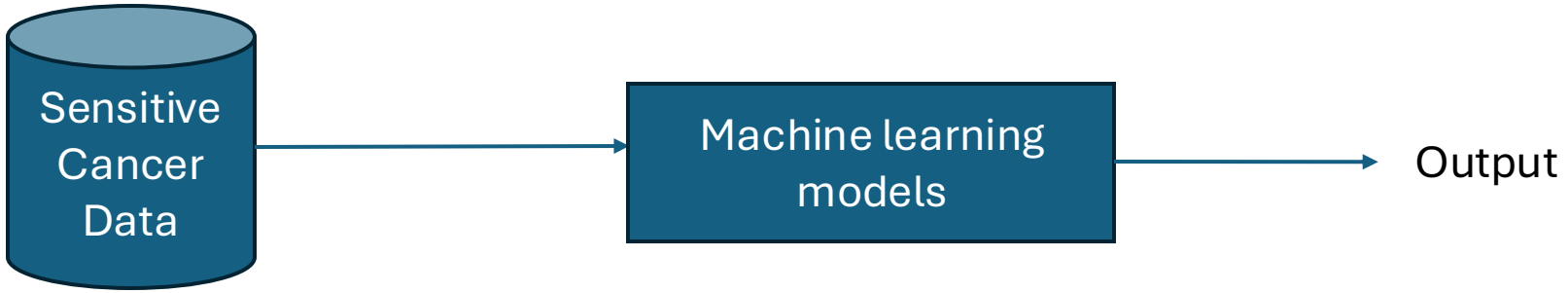
Tasks
Classify the cancer grade/stages
Segment the cancer region in radiographs
Predict the risk of developing a cancer type

Sarcoma is a rare cancer and accounts for approximately 1 % of all diagnosed cancer cases in Europe. In 2023, 570 new cases of sarcoma were recorded in Norway.

Number of available samples in one country may not be sufficient enough for training

How to address the data scarcity of rare cancer types when training machine learning models?

Data Bias Challenge



Tasks
Classify the cancer grade/stages
Segment the cancer region in radiographs
Predict the risk of developing a cancer type

Datasets may contain sampling bias

Melanoma (Skin cancer) diagnosis (images from white skin vs images from dark skin)

Machine learning models trained on biased samples are not suitable for deployment in clinical settings with a diverse population

How to increase the diversity in the datasets?

Faster Data Analysis Challenge

International Agency for Research on Cancer

World Health Organization

NORDCAN

Association of the Nordic Cancer Registries

NORDIC CANCER UNION

The Danish Cancer Registry

Finnish Cancer Registry

Icelandic Cancer Registry

The Cancer Registry of Norway

The Swedish Cancer Registry

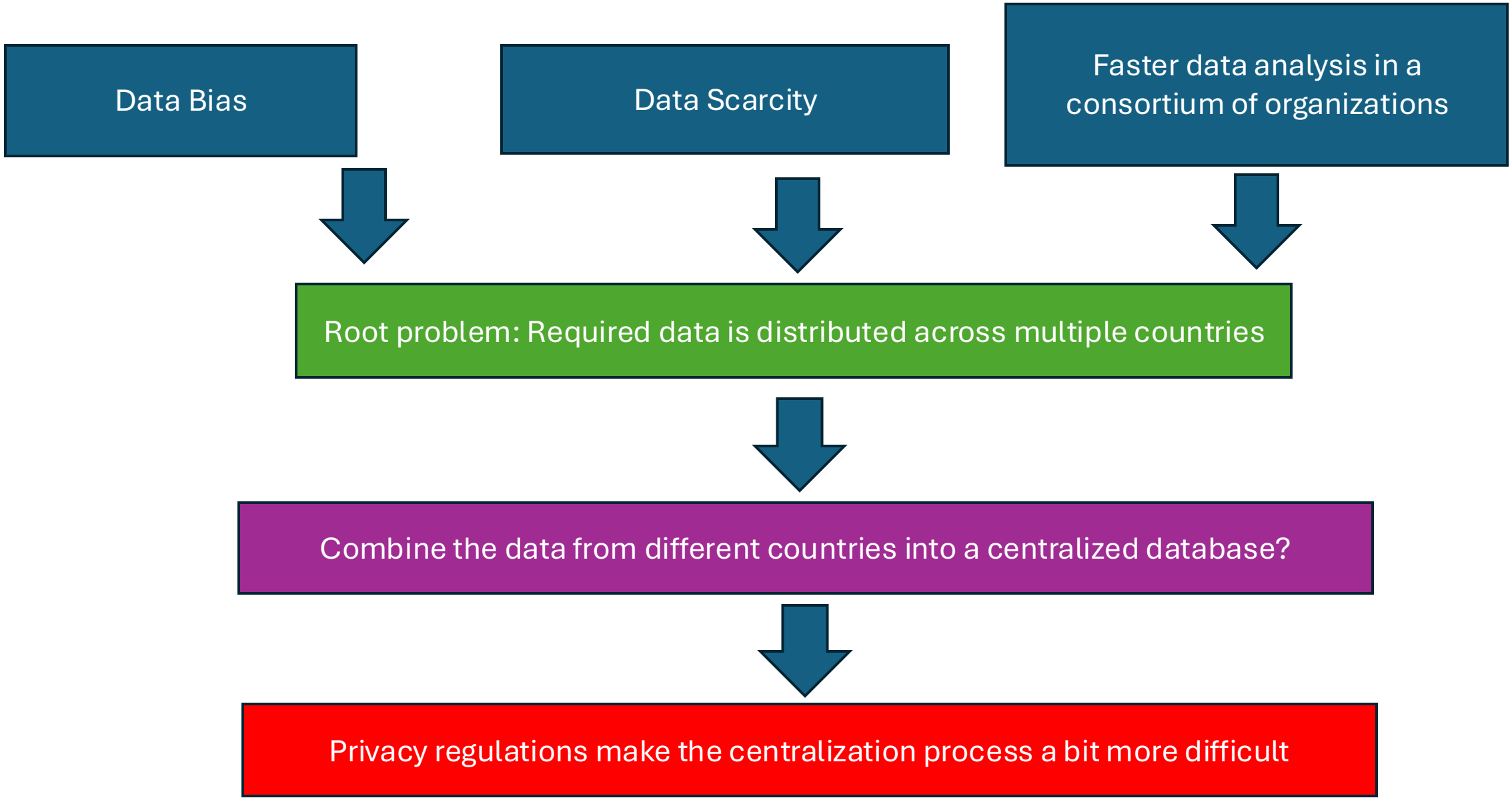
The Faroe Islands Cancer Registry

The National Board of Health of Greenland

Compute comparable cancer statistics for the Nordic countries for long time periods

How can we speed up and improve the process to compute NORDCAN statistics more efficiently than the current approach?

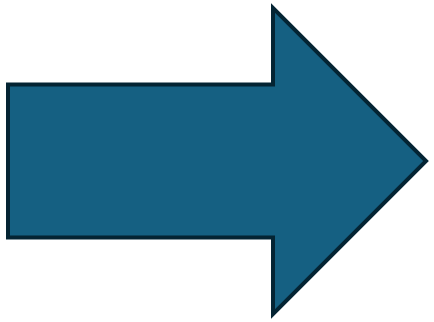
Challenge: Privacy legislation



Motivation from the Cancer Registry Perspective

Challenges

- Data Scarcity
- Data Bias
- Faster data analysis in a consortium of organizations
- Privacy Legislation



Solution Direction

Explore Federated Computing

Introduction to Federated Computing

Data Federation, Federated Computing, General Framework

Data Federation

Definition:

- A data federation F is a federation of local datasets $\{D_1, D_2, \dots, D_m\}$ held by m data owners.

Local Datasets:

- Each local dataset D_i has n_i objects $\{o_1, o_2, \dots, o_{n_i}\}$.
- Each object o_j has k_j attributes $\{x_1, x_2, \dots, x_{k_j}\}$.

Virtual Database:

- The virtual database of this data federation is denoted by the union of these local datasets, i.e.,

$$D = D_1 \cup D_2 \cup \dots \cup D_m$$

Organization 1

Data Owner 1

Dataset 1 (D_1)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$
o_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$

Organization 2

Data Owner 2

Dataset 2 (D_2)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$
o_4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$

Organization 3

Data Owner 3

Dataset 3 (D_3)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_5	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$
o_6	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$

Organization 4

Data Owner 4

Dataset 4 (D_4)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_7	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$
o_8	$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	$x_{8,4}$

Cancer Registry of Norway

Data Owner 1

Dataset 1 (D_1)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$
o_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$

Cancer Registry of Sweden

Data Owner 2

Dataset 2 (D_2)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$
o_4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$

Cancer Registry of Finland

Data Owner 3

Dataset 3 (D_3)

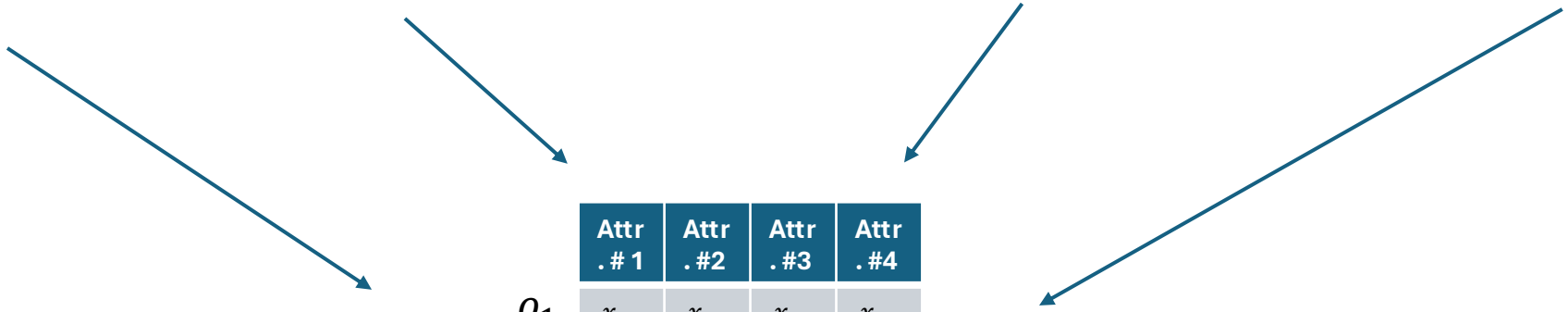
	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_5	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$
o_6	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$

Cancer Registry of Denmark

Data Owner 4

Dataset 4 (D_4)

	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_7	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$
o_8	$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	$x_{8,4}$



	Attr. # 1	Attr. #2	Attr. #3	Attr. #4
o_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$
o_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$
o_3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$
o_4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$
o_5	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$
o_6	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$
o_7	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$
o_8	$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	$x_{8,4}$

Data Federation: Virtual Database

Federated Computing in Data Federations

Federated Computing Objective:

- Compute the result of a task $T(D)$ over the virtual database $D = \bigcup_{i=1}^{m} D_i$ in a data federation F of m data owners $\{D_i\}$.

Federated Computing Key Constraints:

- Autonomous constraint:
  - Each data owner does not share his raw data to anyone.
  - Data owners retain control over their local datasets.
- Security constraint:
  - During the computation, protect against privacy attacks.

Federated Computing Attack Models:

- **Semi Honest adversary**: follows the computation protocol but may try to infer sensitive data
- **Malicious adversary**: deviates from the computation protocol with the intent to infer or expose sensitive data

Federated Analytics vs Federated Learning vs Federated Computing

Fig. 1. Intersection of definitions for FC, FL, and FA. FC consists of FL and FA, whereas FL systems can contain FA characteristics. FA is a subset of FL and FL is a subset of FC.

Federated Computing - Survey on Building Blocks, Extensions and Systems

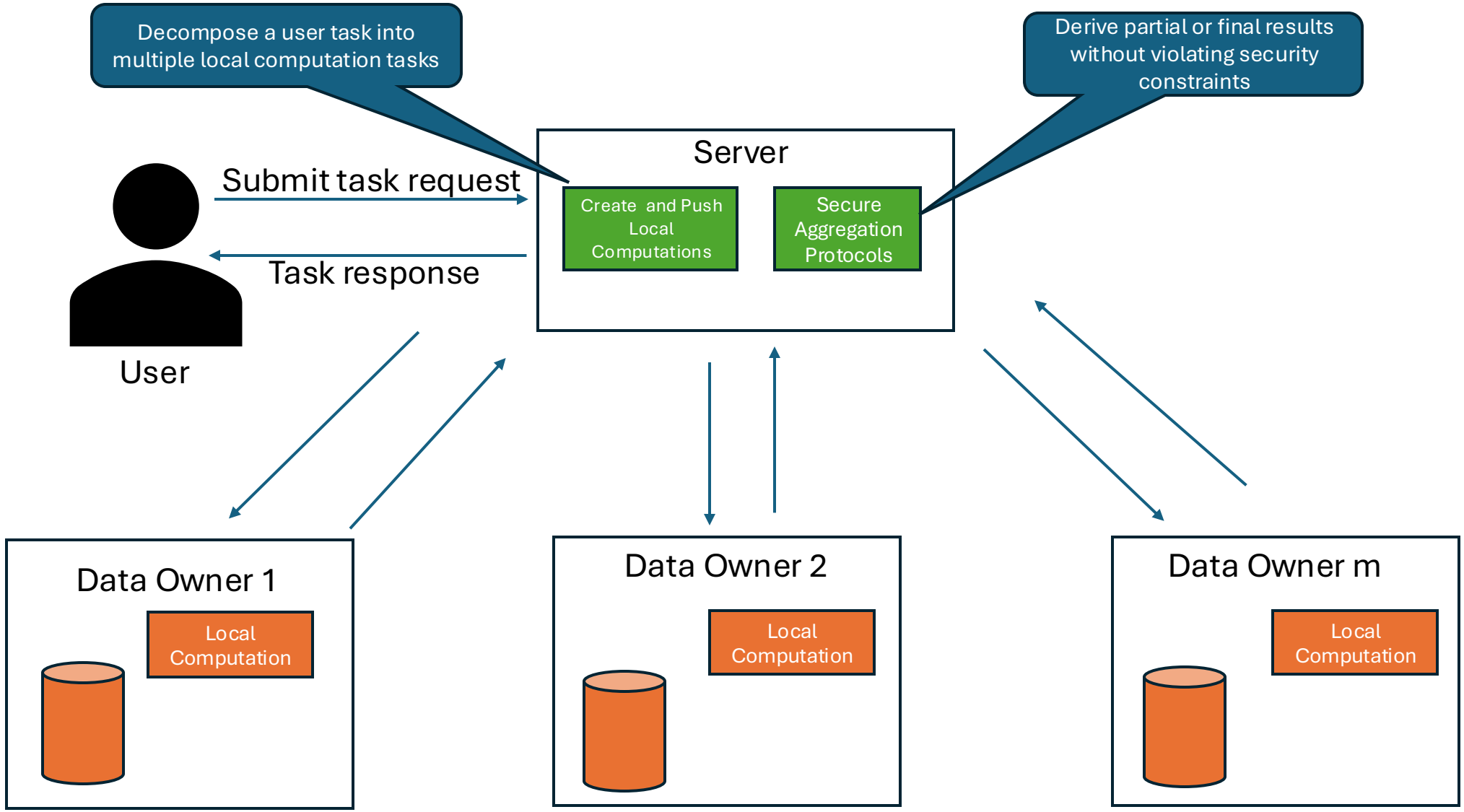
RENÉ SCHWERMER, Technical University of Munich, Germany
RUBEN MAYER, University of Bayreuth, Germany
HANS-ARNO JACOBSEN, University of Toronto, Canada

Federated Computing: Query, Learning, and Beyond

Yongxin Tong[†] Yuxiang Zeng[†,‡] Zimu Zhou[♯] Boyi Liu[†] Yexuan Shi[†]
Shuyuan Li[†] Ke Xu[†] Weifeng Lv[†]
[†] State Key Laboratory of Software Development Environment,
Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing,
School of Computer Science, Beihang University, Beijing, China
{yxtong,turf1013,liuby,skyxuan,lishuyuan,kexu,lwf}@buaa.edu.cn
[‡] The Hong Kong University of Science and Technology, Hong Kong SAR, China
[♯] City University of Hong Kong, Hong Kong SAR, China zimuzhou@cityu.edu.hk

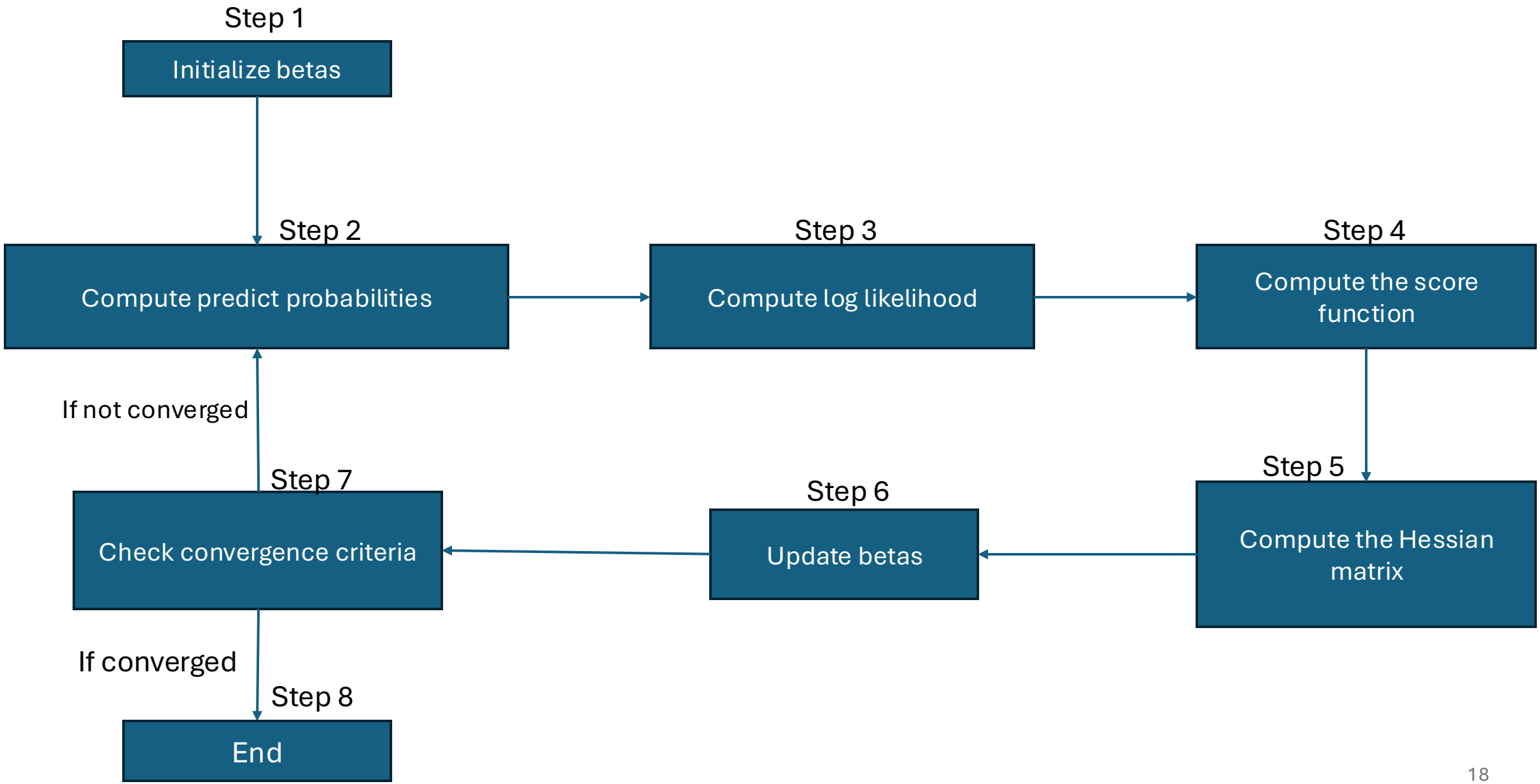
Refer to the literature to know more differences between Federated Analytics and Federated Learning

General Framework for Federated Computing Tasks

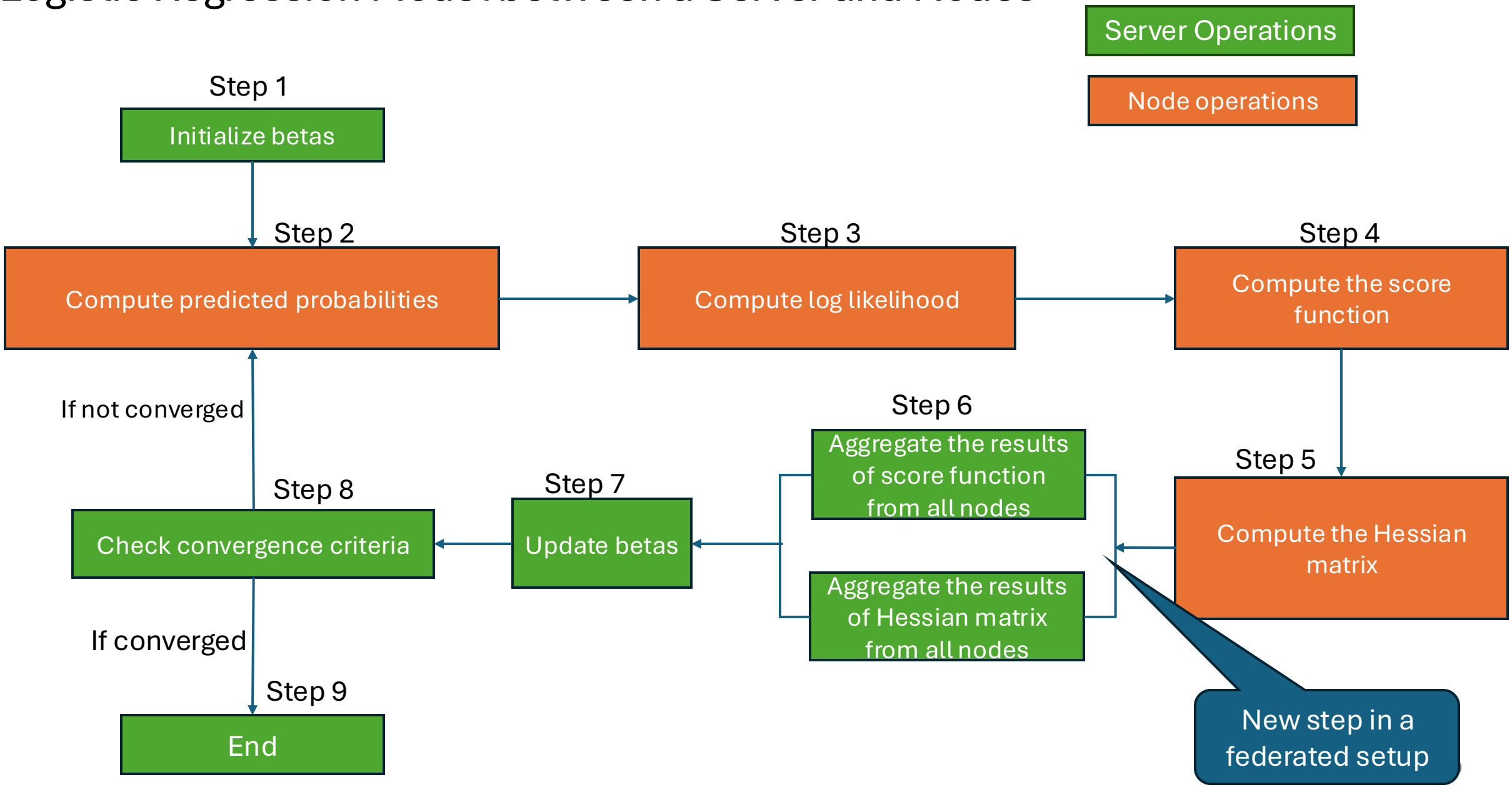


A Federated Learning Example

Centralized Workflow of Maximum Likelihood Estimation of logistic regression models



Federated Learning: Division of Maximum Likelihood Estimation Operations of Logistic Regression Model between a Server and Nodes



A mathematical view of federated logistic regression models

Server Operations

Generate and send initial β

Receive $U_j(\beta), H_j(\beta)$ $j \in M$

Aggregate operations

$$U(\beta_{old}) = \frac{1}{|M|}\sum_{m=1}^{|M|} U_m(\beta)$$

$$H(\beta_{old}) = \frac{1}{|M|}\sum_{m=1}^{|M|} H_m(\beta)$$

Update model parameters

$$\beta_{new} = \beta_{old} - H^{-1}(\beta_{old}) \cdot U(\beta_{old})$$

Convergence Criteria

$$\| \beta_{new} - \beta_{old} \| < \epsilon$$

Send β_{new} or converged message to each data owner

Data Owner 1 (Node 1)

Receive β, β_{new}

$$P_i = \sigma(\beta^T x_i) = \frac{1}{1 + e^{-\beta^T x_i}}$$

$$U_1(\beta) = \sum_i (y_i - P_i) x_i$$

$$H_1(\beta) = -\sum_i (P_i(1 - P_i) x_i x_i^T$$

Send $U_1(\beta), H_1(\beta)$

Data Owner 2 (Node 2)

Receive β, β_{new}

$$P_i = \sigma(\beta^T x_i) = \frac{1}{1 + e^{-\beta^T x_i}}$$

$$U_2(\beta) = \sum_i (y_i - P_i) x_i$$

$$H_2(\beta) = -\sum_i (P_i(1 - P_i) x_i x_i^T$$

Send $U_2(\beta), H_2(\beta)$

Data Owner n (Node n)

Receive β, β_{new}

$$P_i = \sigma(\beta^T x_i) = \frac{1}{1 + e^{-\beta^T x_i}}$$

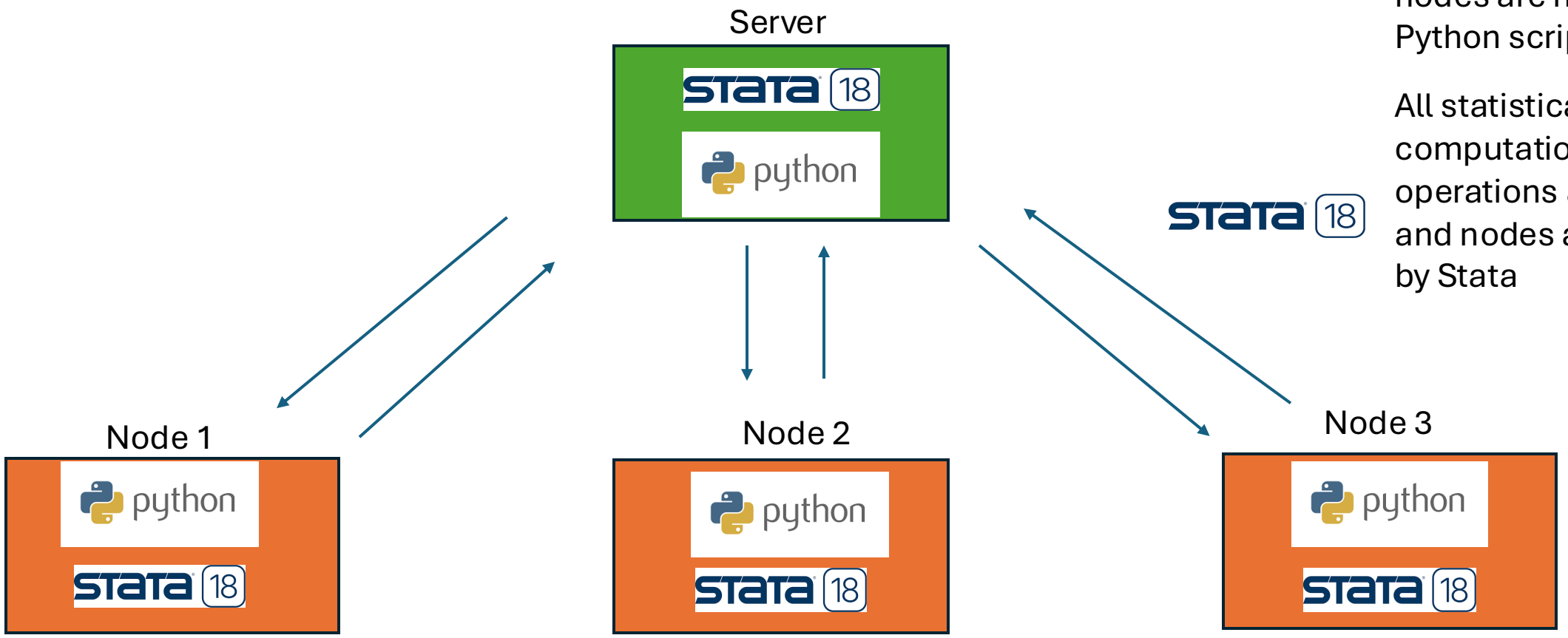
$$U_n(\beta) = \sum_i (y_i - P_i) x_i$$

$$H_n(\beta) = -\sum_i (P_i(1 - P_i) x_i x_i^T$$

Send $U_n(\beta), H_n(\beta)$

Integrating Federated Computing with Stata

Option 1a: Leverage existing capabilities



All message passing communication between server and nodes are handled by Python scripts

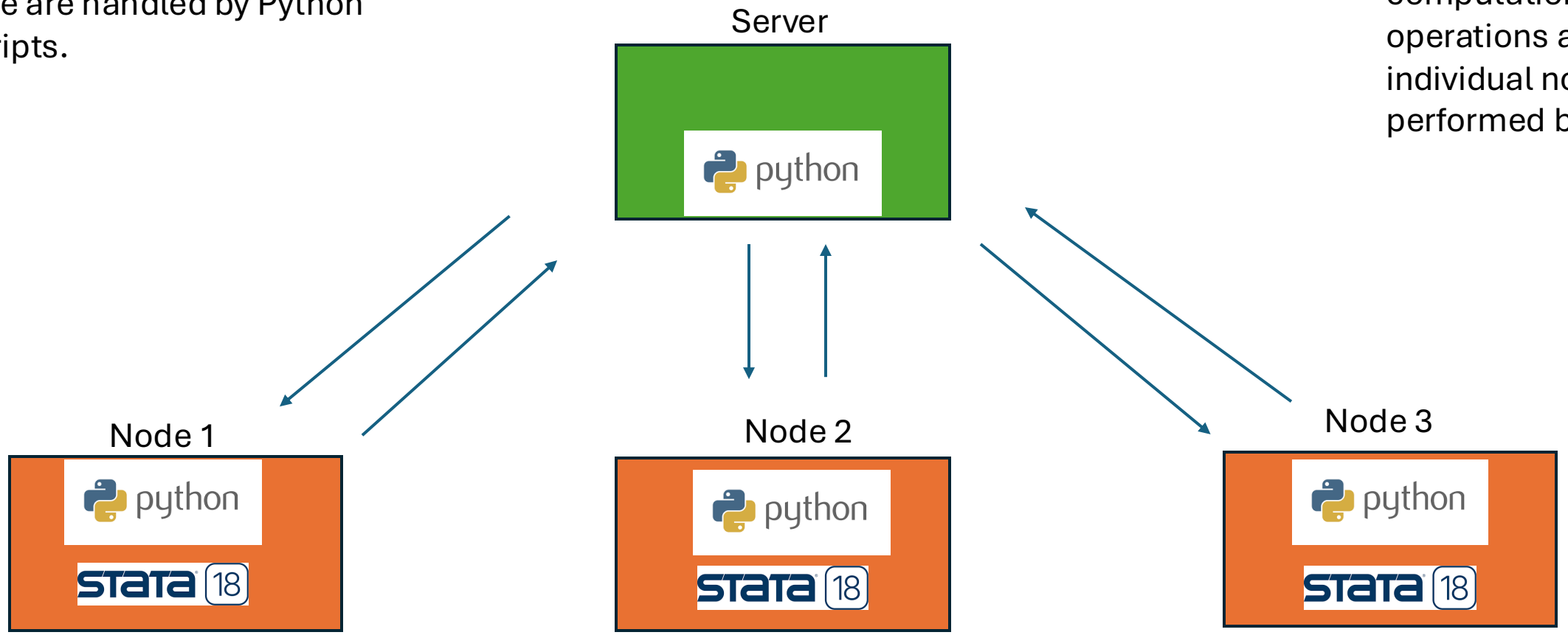
All statistical computation operations at Server and nodes are done by Stata

Stata's Python integration capabilities are utilized to exchange information between Python and Stata scripts.

Option 1b: Leverage existing capabilities

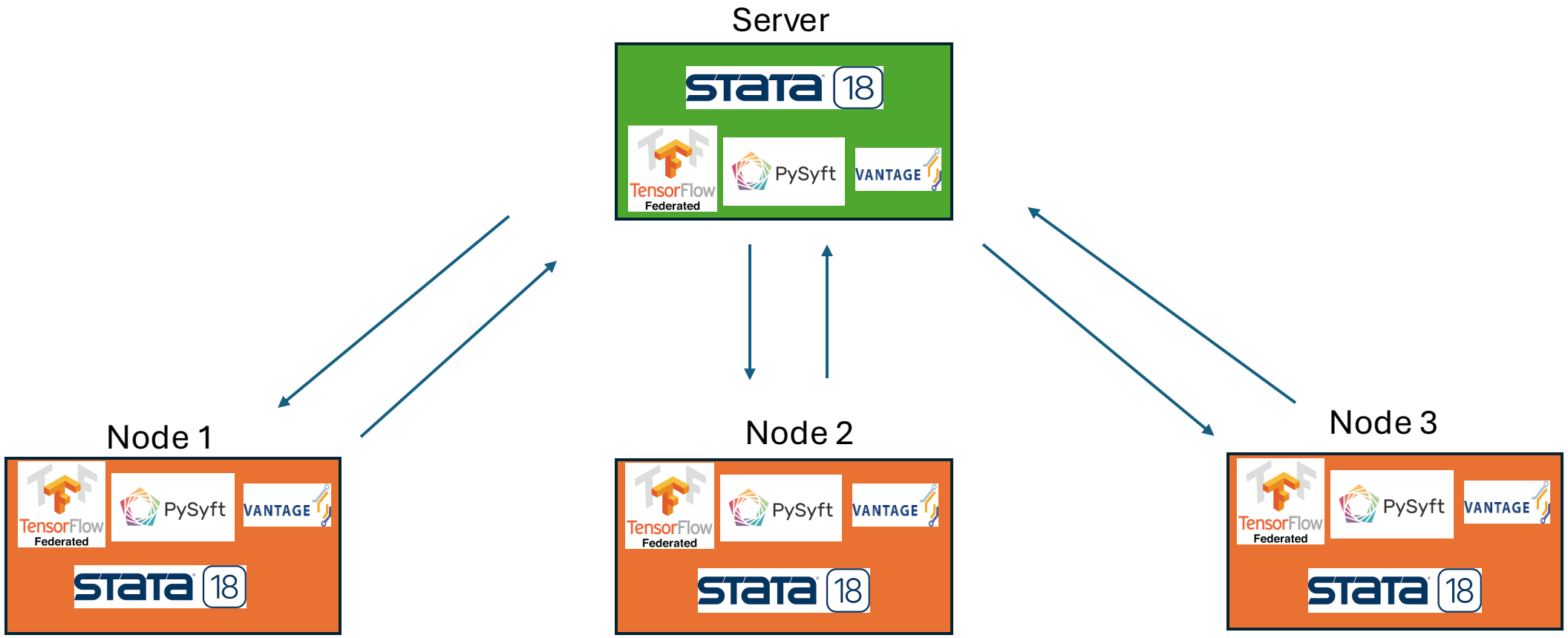
- All operations at the server side are handled by Python scripts.

All statistical computation operations at individual nodes are performed by Stata



Stata's Python integration capabilities are utilized to exchange information between Python and Stata scripts.

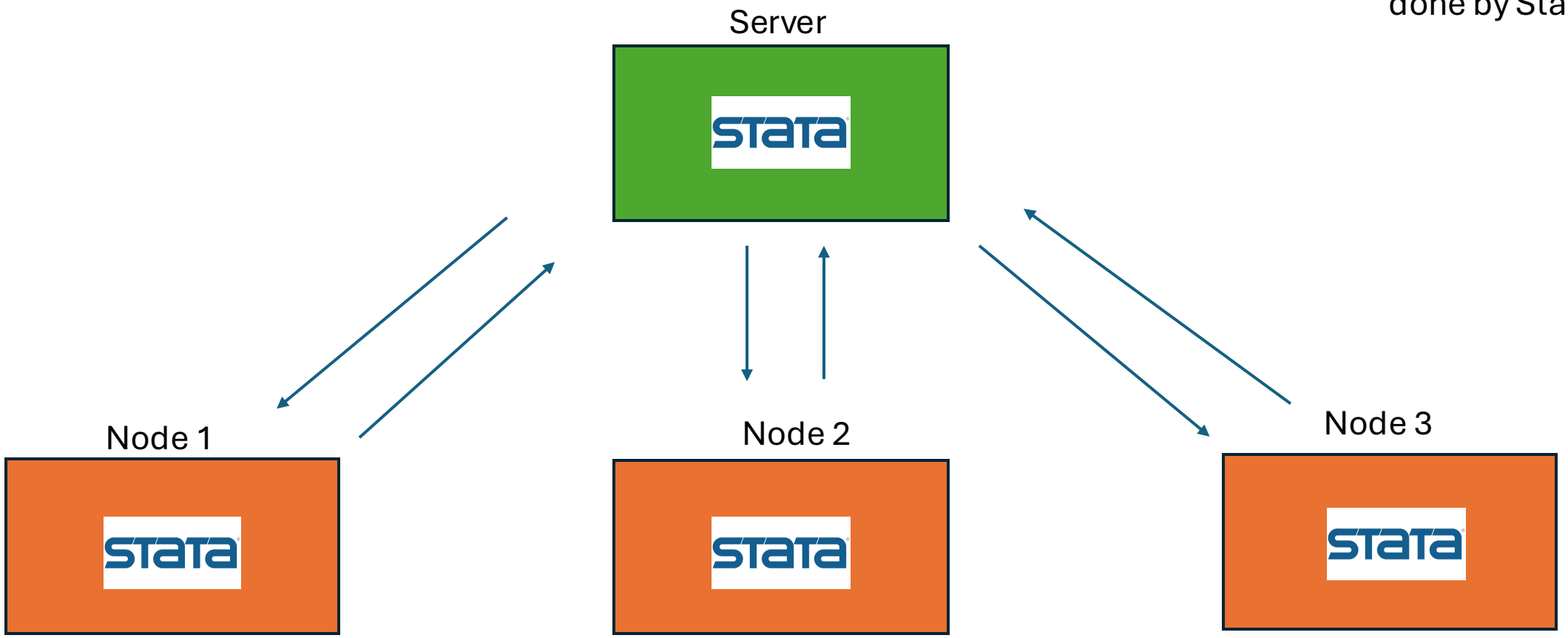
Option 2: Integrate with one of the existing federated computing frameworks



Stata's Python integration capabilities are utilized to integrate with appropriate Federated Computing frameworks .

Option 3: Native support

All operations at Server and nodes are done by Stata



StataCorp creates a new build with a native support to federated computing in the future.

Rudimentary comparison between the three options

Criteria	Python Script Communication with Stata	Integrate with Existing Federated Computing Framework	Provide Native Support in Stata
Ease of Integration	Moderate (Requires custom Python scripts to bridge between Stata and federated nodes)	High (Existing frameworks likely have better support and documentation)	Low (Native support would require Stata to develop complex features)
Performance	Moderate (Overhead from Python Stata Communication)	High? (Optimized frameworks may offer better performance)	Very High (Direct, optimized native support)
Flexibility	High (Python allows customization and flexibility in handling specific use cases)	Moderate (Frameworks may allow some customization but are more structured)	Low (Native implementation may be rigid and less customizable)
Maintenance	High (Python scripts require continuous maintenance and compatibility updates)	High (Maintenance is required for both the FL framework and Python Stata bridge)	Low (Native support once implement may require fewer updates)
Security	Moderate (Python adds another layer of complexity and possible vulnerability)	Moderate (Existing frameworks are often optimized for security but still adds complexity)	High (Native support could have optimized security features)
Compatibility with Stata Workflows	Moderate (Python communication with Stata adds complexity to existing workflows)	Moderate (FL frameworks need to communicate through Python or other languages, adding complexity)	Very High (Seamless integration with Stata's existing feature and workflows)

Practical Challenges and Limitations in deploying federated computing systems

Preparing the data for the federated computing is consuming more time

Agree on variables to be used	Agree on variable names and formats	Data format standardisation

Collaboration with a group of experts in different areas are required; terminologies used by experts in various domains might be confusing.

Domain data expert

Software Engineer

Data Scientist/ Statistician

Infrastructure Engineer

Researchers/ Use case owners

Legal

Lessons Learned from Deploying Federated Computing Nodes in Cross-Silo Healthcare Settings: Case Studies from the Cancer Registry of Norway

Narasimha Raghavan Veeraragavan, Steinar Auensen, Daan Knoors, Jan F. Nygård
Cancer Registry of Norway
Norwegian Institute of Public Health
Oslo, Norway

Federated computing research is still maturing

Table 1: Comparison of existing systems for federated queries

System	Data Type	Data Integration Mode	Attacker Model	Data Size	#(Data Owner)
SMCQL [2]	Relational	Horizontal	Semi-honest	≤1K	≤2
ShrinkWrap [12]	Relational	Horizontal	Semi-honest	≤40K	≤2
SAQE [13]	Relational	Horizontal	Semi-honest	≤500K	≤2
Conclave [14]	Relational	Horizontal/Vertical	Semi-honest	≤1B	≤3
Hu-Fu [4]	Spatial	Horizontal	Semi-honest	≤1B	≤10
Senate [35]	Relational	Horizontal	Malicious	≤160K	≤16
Opaque [39]	Relational	Vertical	Malicious	≤1M	≤5

Federated Computing: Query, Learning, and Beyond

Yongxin Tong† Yuxiang Zeng†,‡ Zimu Zhou♯ Boyi Liu† Yexuan Shi†
Shuyuan Li† Ke Xu† Weifeng Lv†
† State Key Laboratory of Software Development Environment,
Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing,
School of Computer Science, Beihang University, Beijing, China
{yxtong,turf1013,liuby,skyxuan,lishuyuan,kexu,lwf}@buaa.edu.cn
‡ The Hong Kong University of Science and Technology, Hong Kong SAR, China
♯ City University of Hong Kong, Hong Kong SAR, China zimuzhou@cityu.edu.hk

Framework	Secure Aggregation	Encrypted Training	Differential Privacy	Poisoning Defenses	Attacks Simulation	Traceability	Supply Chain Security
TensorflowFederated	✓	×	✓	×	×	×	×
PySyft	✓	✓	✓	×	×	×	×
FATE	✓	✓	×	×	×	×	×
PaddleFL	✓	×	✓	×	×	✓	×
FedBioMed	✓	×	✓	×	×	×	×
Substra	×	×	✓	×	×	✓	×
FedML	✓	×	✓	✓	✓	×	×
FLower	✓	×	✓	×	×	×	×
FederatedScope	✓	✓	✓	✓	✓	×	×
OpenFL	×	×	✓	×	×	×	×
NVFLARE	✓	×	✓	×	×	×	×
APPFL	×	×	✓	×	×	×	×
Vantage6	×	×	×	×	×	×	×
FedN	×	×	×	×	×	×	×

Conferences > 2023 Eighth International Con...

Federated Learning Showdown: The Comparative Analysis of Federated Learning Frameworks

Publisher: IEEE Cite This PDF

Sai Praneeth Karimireddy ; Narasimha Raghavan Veeraragavan ; Severin Elvatun ; Jan F. Nygård All Authors

Opensource frameworks are available.

Understand the framework's specification and check if it can support your use case

Demo
(Stata Simulation of Federated Maximum Likelihood Estimation of Logistic Regression Models)

Stata Example

- Use Stata `lbw.dta`
  - Randomly split data into 3 separate data sets.
  - Save each data set to a separate folder (representing 3 nodes).
  - Main Stata (server) session initiates 3 separate Stata sessions.

Main (server) Stata session

```
// Start stata session in each node
forvalues node = 1/3 {
  winexec S:\Prog64\Stata\Stata18MP/StataMP-64.exe /e                    ///
          do ${root}/data/Node`node'/node_setup`node'.do `node' ${root}
}

// Fit Model
ml model d2 logistic_master ///
   (xb: low = age smoke),   ///
   search(off) maximize
ml display
```

- Needs simulated data in memory with the correct N and variable names
- Use method `d2` as `ml` requires that only aggregated information is returned
- Methods `lf2` and `gf2` both require individual level information returned to `ml`.

Key ado/do files

logistic_master.ado

```
// Send (copy) beta vector to each node folder
// Wait for all updated log-likelihood, gradient and Hessian
// Sum likelihood, gradient and Hessian contributions from each node
// Return likelihood, gradient and Hessian to ml to update beta vector
```

logistic_node.do

```
// Wait for updated beta matrix
// Calculate log-likelihood, gradient and Hessian
// Send (copy) to main server folder
```

```
. ml model d2 logistic_master ///
>     (xb: low = age smoke), search(off) maximize

Main server:
  -- Copying beta matrix and node_instructions.do to each node
  -- Node 1
       -- Waiting for likelihood, gradient and Hessian.
       -- Reading in likelihood, gradient and Hessian.
  -- Node 2
       -- Waiting for likelihood, gradient and Hessian.
       -- Reading in likelihood, gradient and Hessian.
  -- Node 3
       -- Waiting for likelihood, gradient and Hessian.
       -- Reading in likelihood, gradient and Hessian.
  -- Summing contributions over Nodes.
  -- Returning likelihood, gradient and Hessian to ml.
Iteration 0:  Log likelihood = -131.00482
.
.
.
```

```
Main server:
  -- Copying beta matrix and node_instructions.do to each node
  -- Node 1
      -- Waiting for likelihood, gradient and Hessian.
      -- Reading in likelihood, gradient and Hessian.
  -- Node 2
      -- Waiting for likelihood, gradient and Hessian.
      -- Reading in likelihood, gradient and Hessian.
  -- Node 3
      -- Waiting for likelihood, gradient and Hessian.
      -- Reading in likelihood, gradient and Hessian.
  -- Summing contributions over Nodes.
  -- Returning likelihood, gradient and Hessian to ml.
Iteration 3:  Log likelihood = -113.63815

. ml display, noheader
------------------------------------------------------------------------------
         low | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |  -.0497793    .031972    -1.56   0.119    -.1124432    .0128846
       smoke |   .6918487   .3218061     2.15   0.032     .0611203    1.322577
       _cons |   .0609055   .7573199     0.08   0.936    -1.423414    1.545225
------------------------------------------------------------------------------
```

```
. est tab main_server nodes, eq(1) se modelwidth(11)

-----------------------------------------
    Variable | main_server      nodes     
-------------+---------------------------
         age |  -.04977925    -.04977927  
             |   .03197195     .03197196  
       smoke |    .6918486     .69184867  
             |   .32180611     .32180612  
       _cons |    .0609051     .06090554  
             |   .75731987      .7573199  
-----------------------------------------
                            Legend: b/se
```

- Same parameter estimates with and without having data in Main Server Stata session

Summary

- Federated computing leverages the power of distributed datasets while helping to comply with privacy regulations.
- Research on federated computing is continuously growing.
- The integration with Stata opens new opportunities for secure, efficient and collaborative data analysis in various fields.