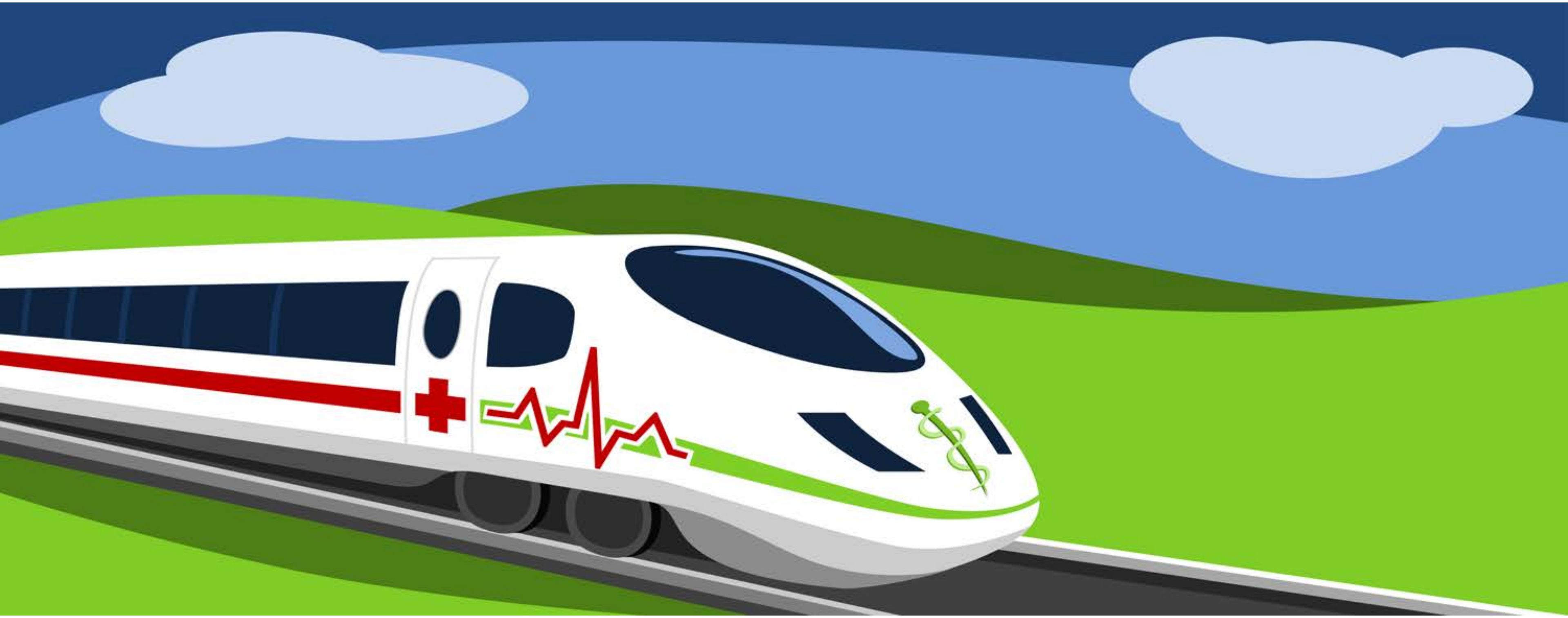
MEDIZIN INFORMATIK INITIATIVE

DIFUTURE

Data Integration for Future Medicine

Universitätsklinikum Tübingen

EBERHARD KARLS UNIVERSITÄT TÜBINGEN



Personal Health Trains

A Status Update

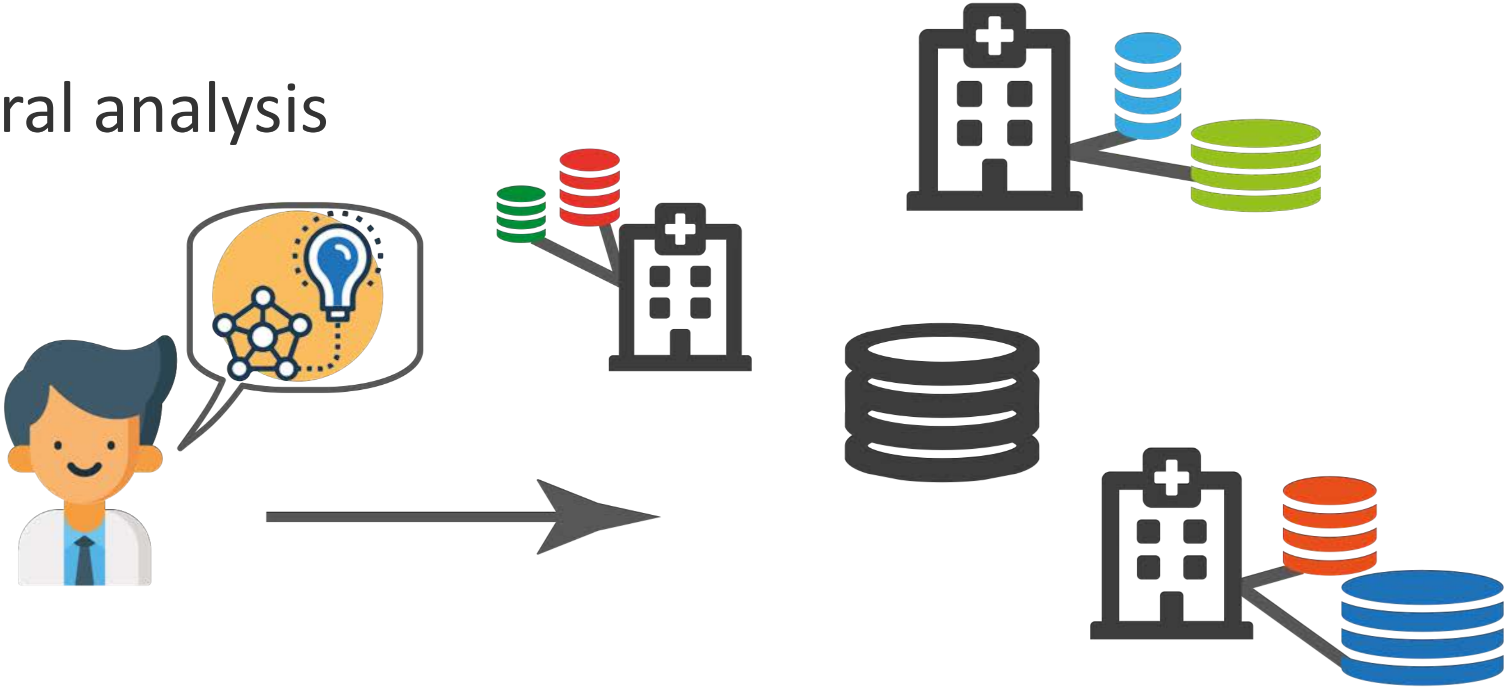
Marius Herr, Lukas Zimmermann, Mete Akgün, Nico Pfeifer, Oliver Kohlbacher

University of Tübingen & University Hospital Tübingen

PfeiferLab.org & KohlbacherLab.org

Background PHT

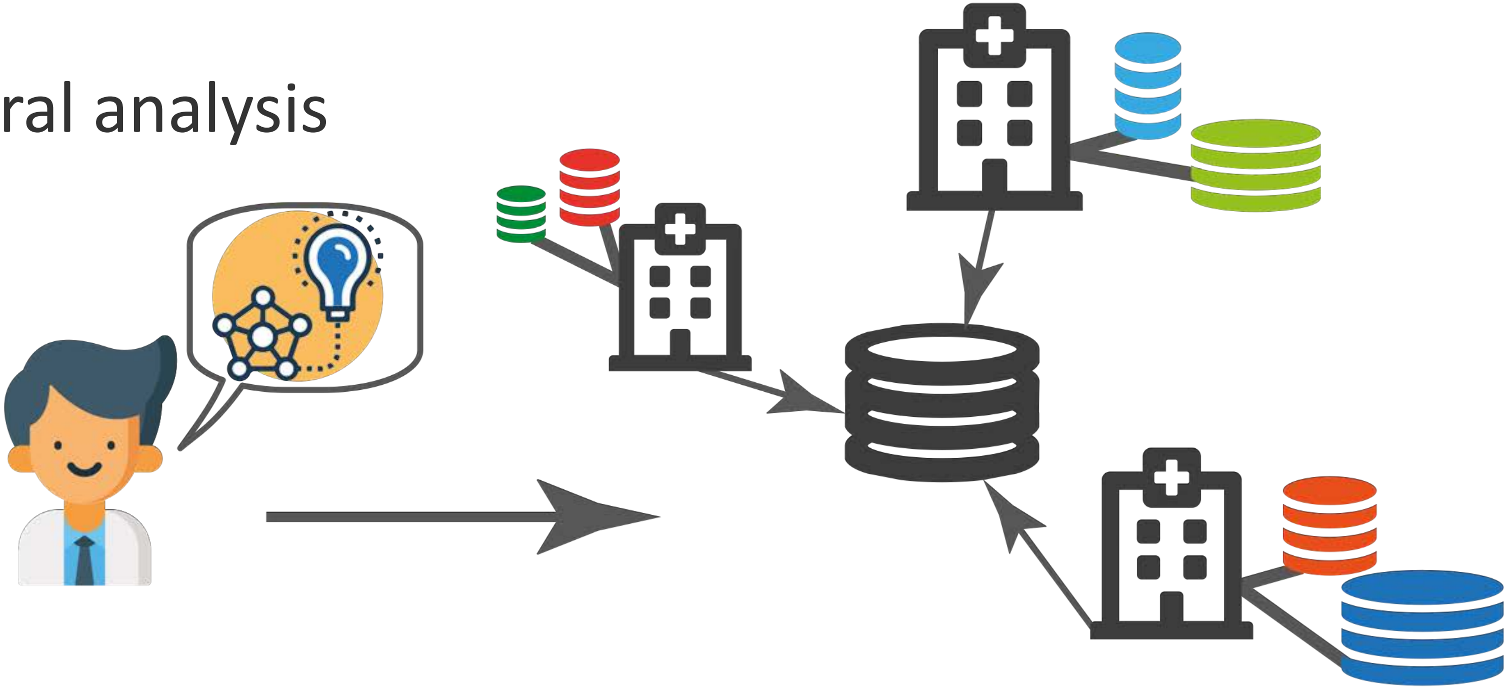
Central analysis



Researcher wants to do an analysis across hospitals

Background PHT

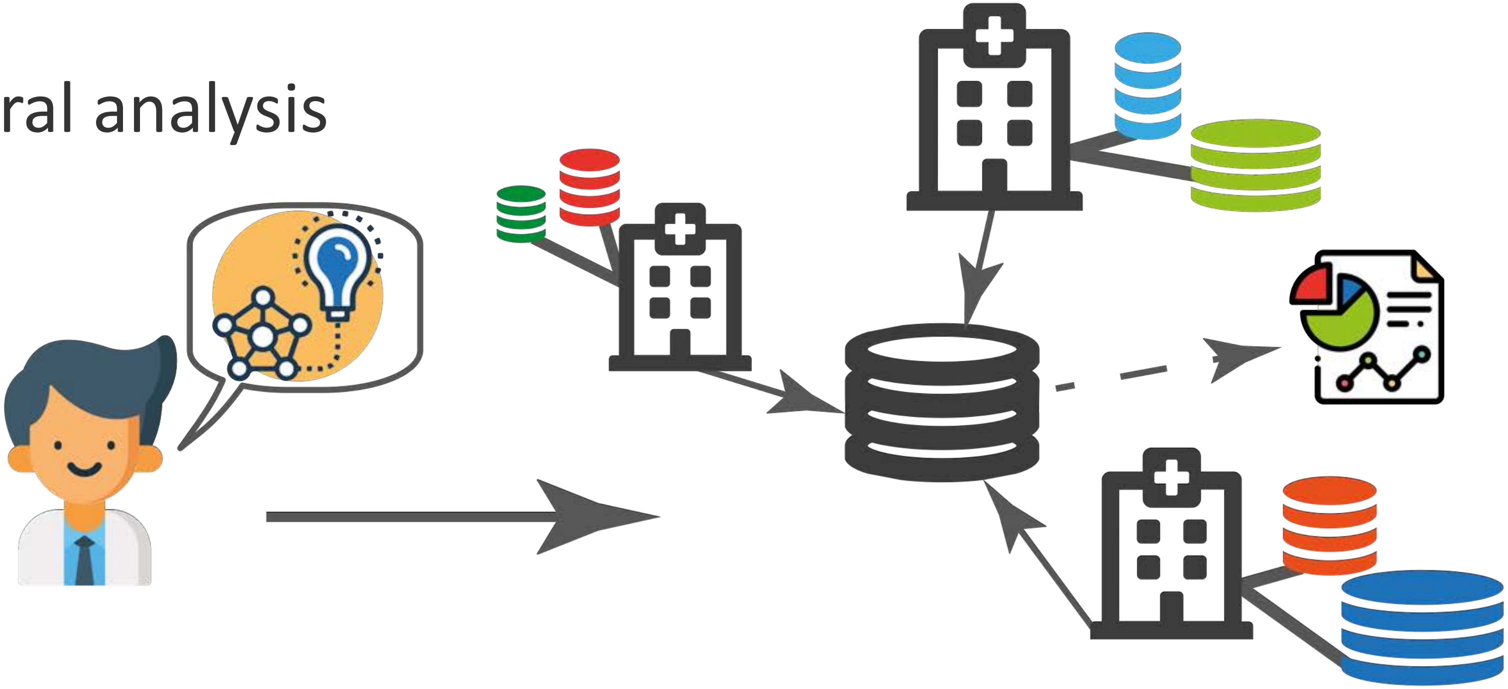
Central analysis



All sites send data to computation place

Background PHT

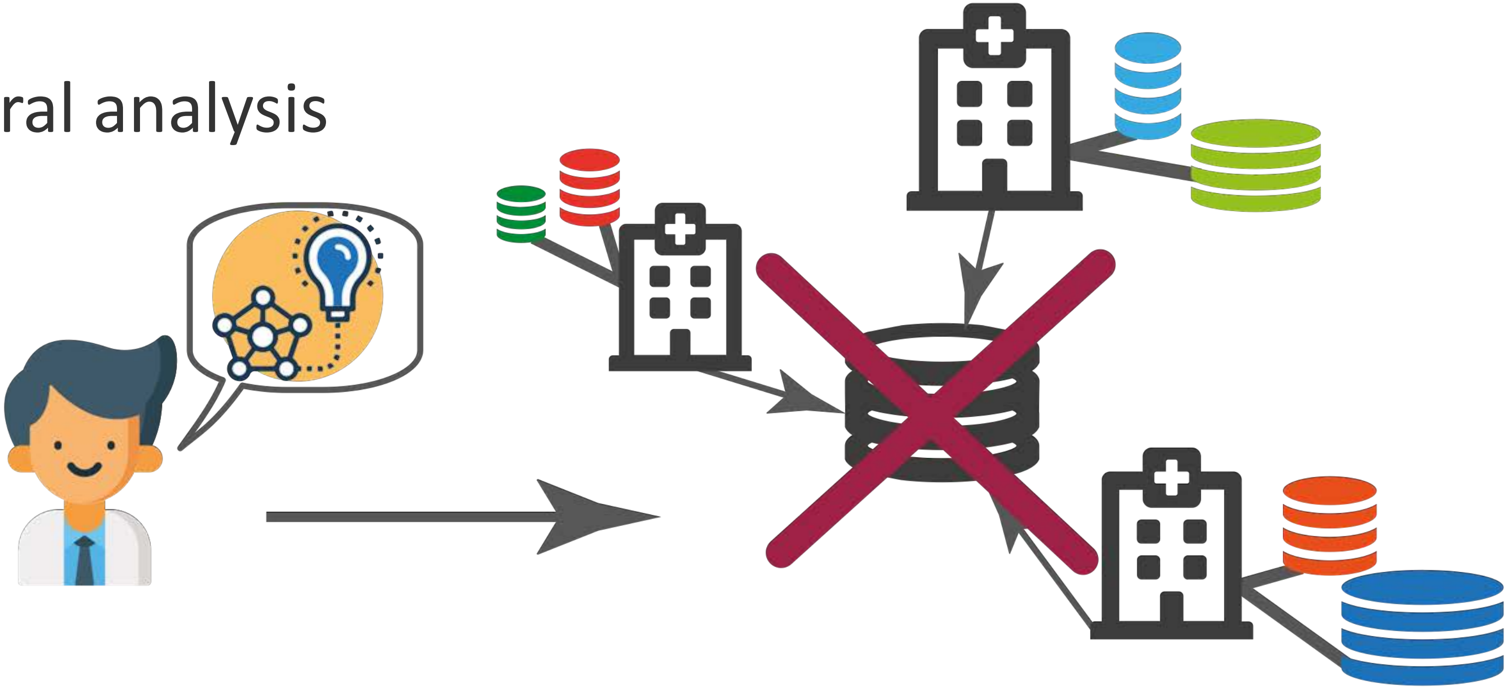
Central analysis



Researcher creates model and obtains results

Background PHT

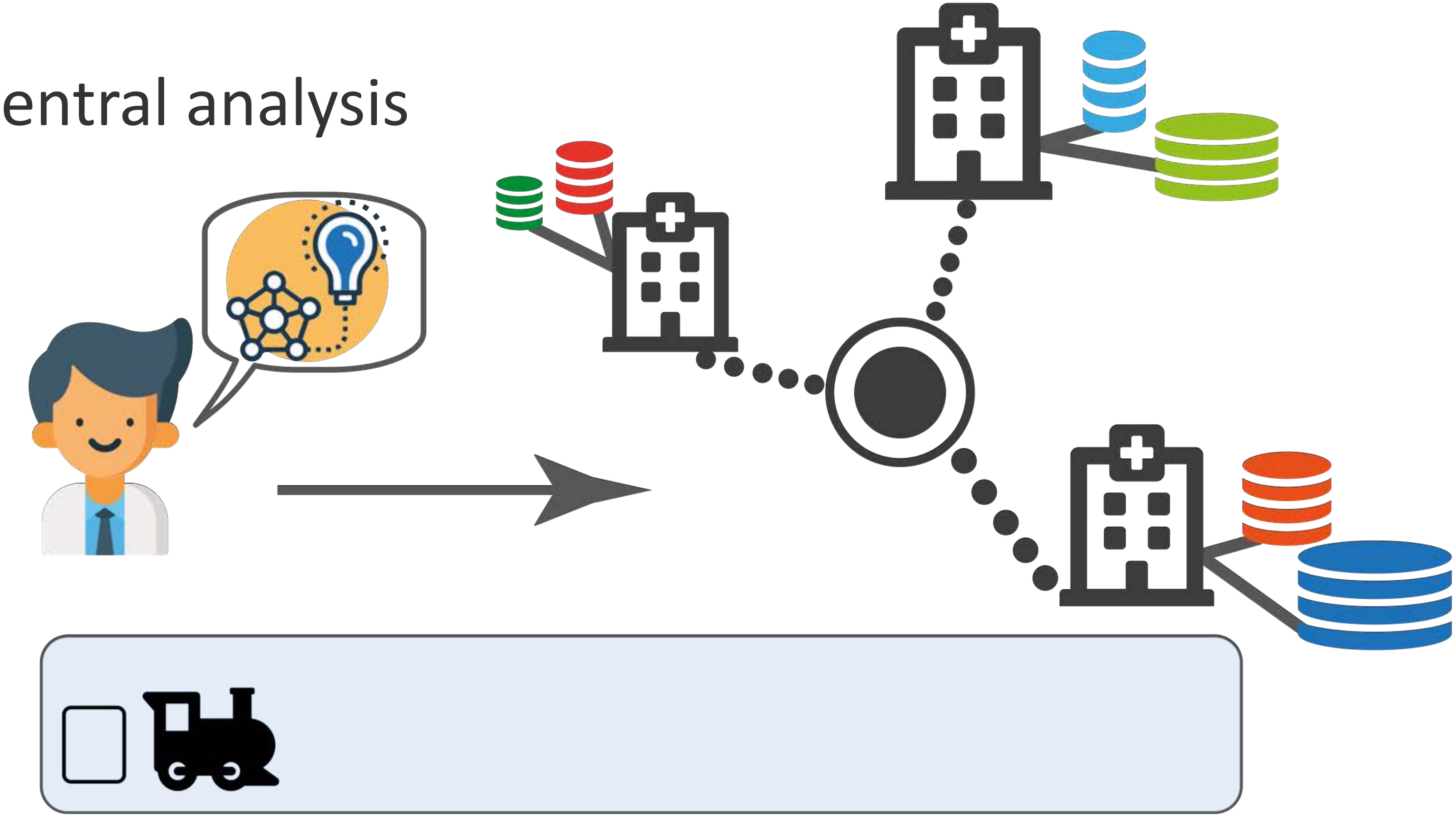
Central analysis



Sites loose control over data!

Background PHT

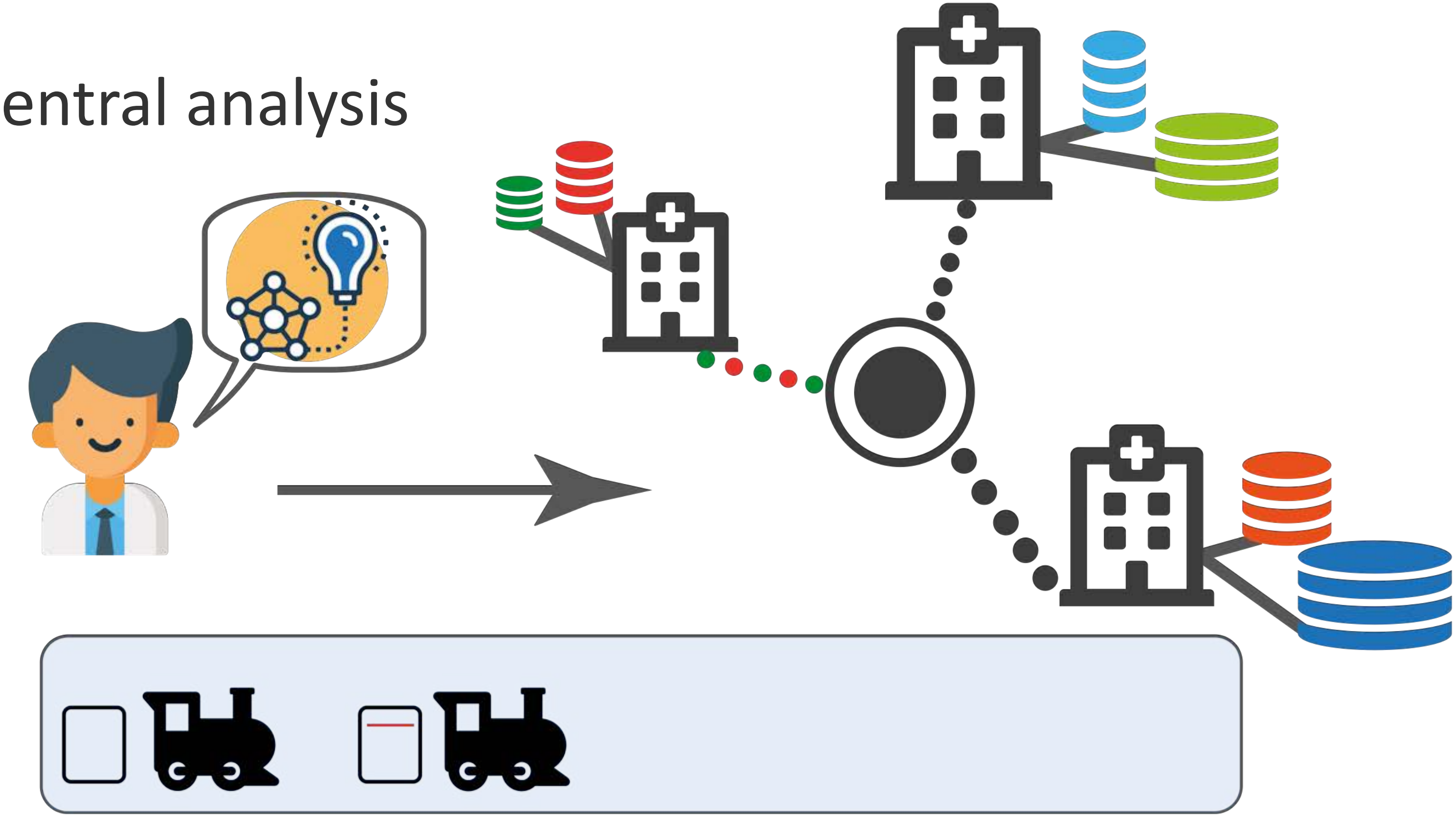
Decentral analysis



***Key idea:** "Bring algorithms to health data, instead of bringing data to algorithms."*

Background PHT

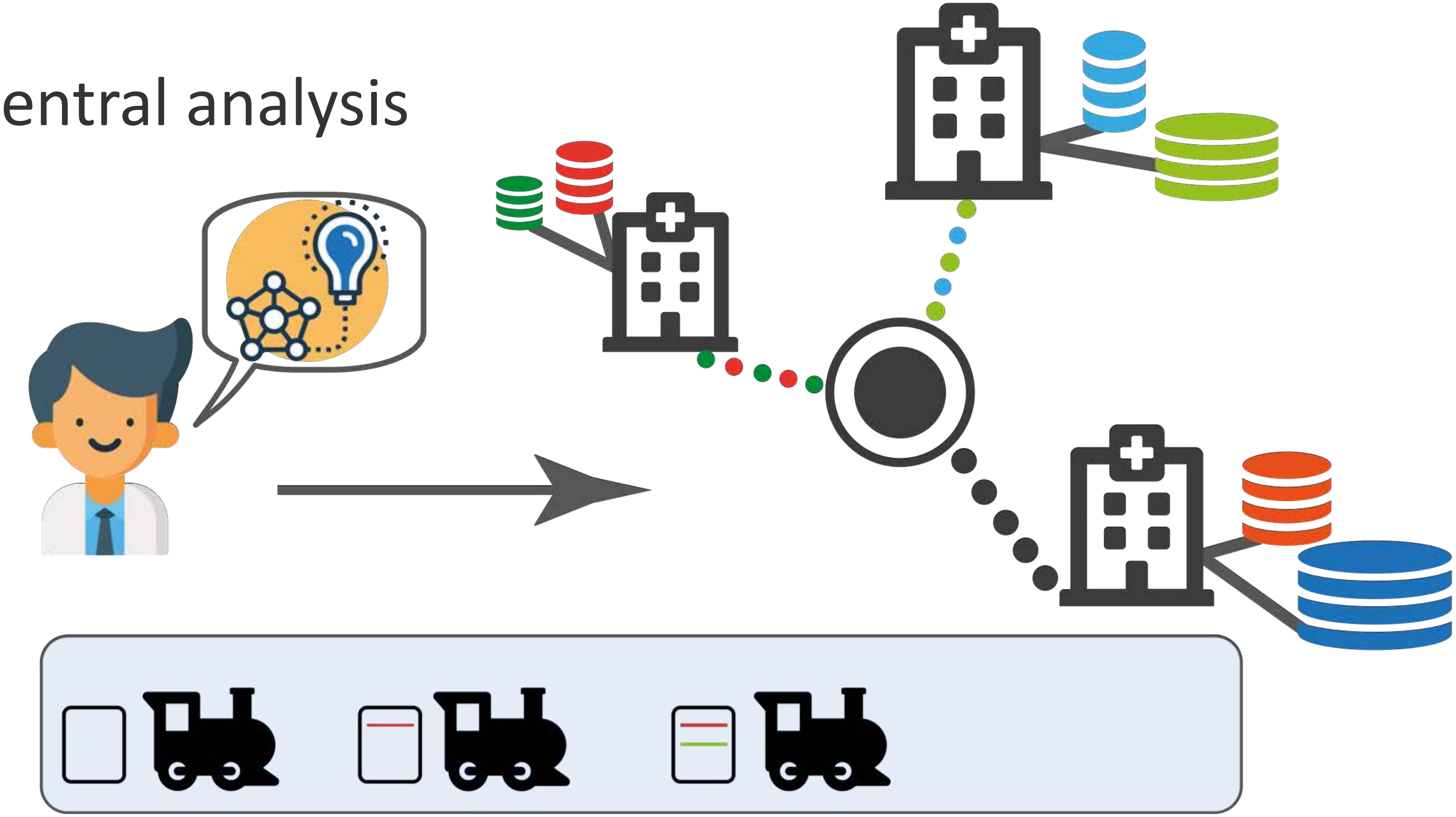
Decentral analysis



Online Learning: Train model on first site

Background PHT

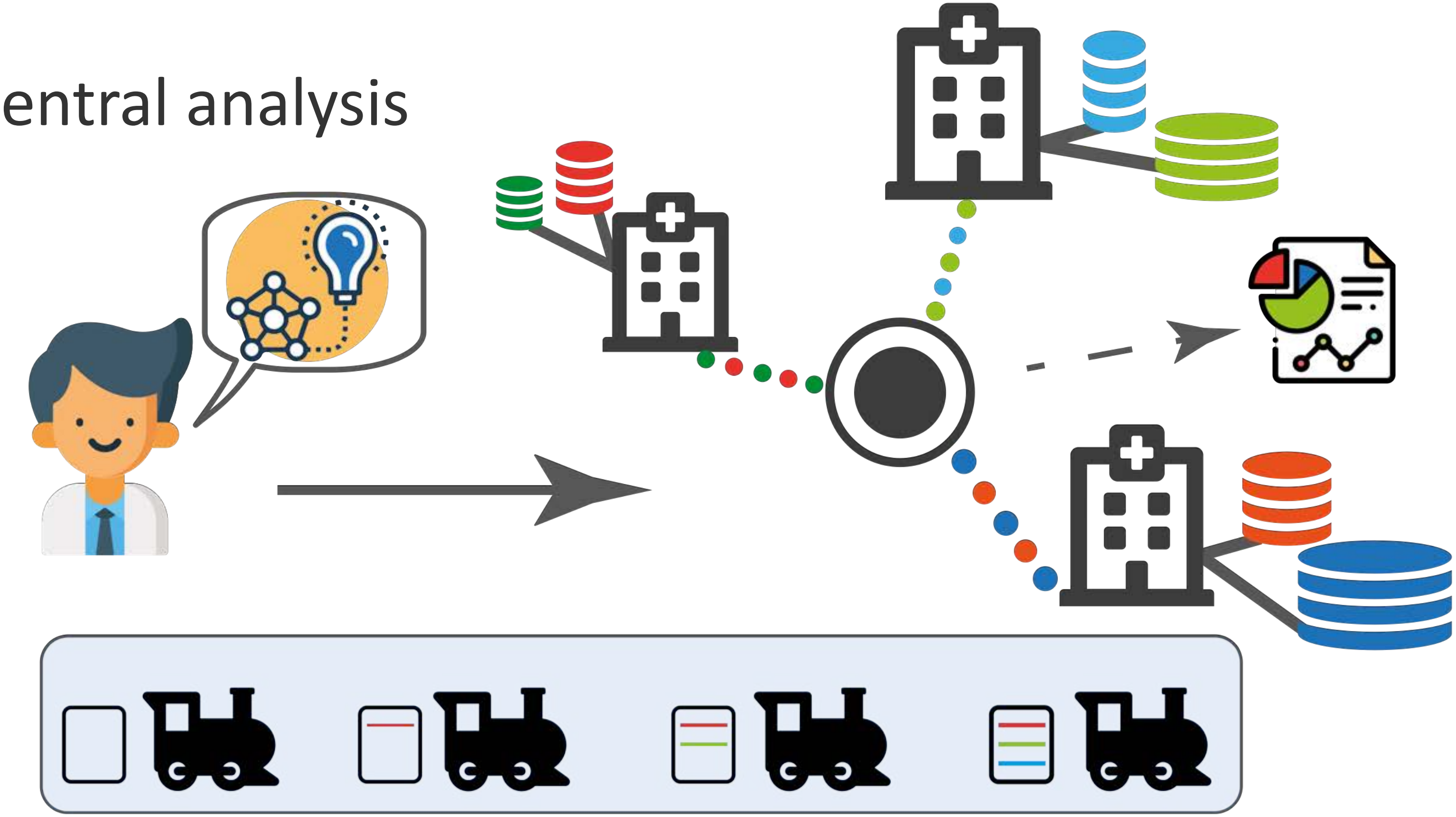
Decentral analysis



Online Learning: Update model on second site

Background PHT

Decentral analysis



Online Learning: **Finalize model and obtain results**

Background PHT

DTL
DUTCH TECHCENTRE FOR LIFE SCIENCES

Manifesto[1] of PHT from DTL

- Advance healthcare and biomedical science through shared infrastructure
- Keep control over data at each local site
- Machine-readability at the core
- Advance data analysis and medical decision making

[1] https://www.dtls.nl/wp-content/uploads/2017/12/PHT_Manifesto.pdf

Implementation Network

Personal Health Train

https://www.go-fair.org/implementation-networks/overview/personal-health-train/

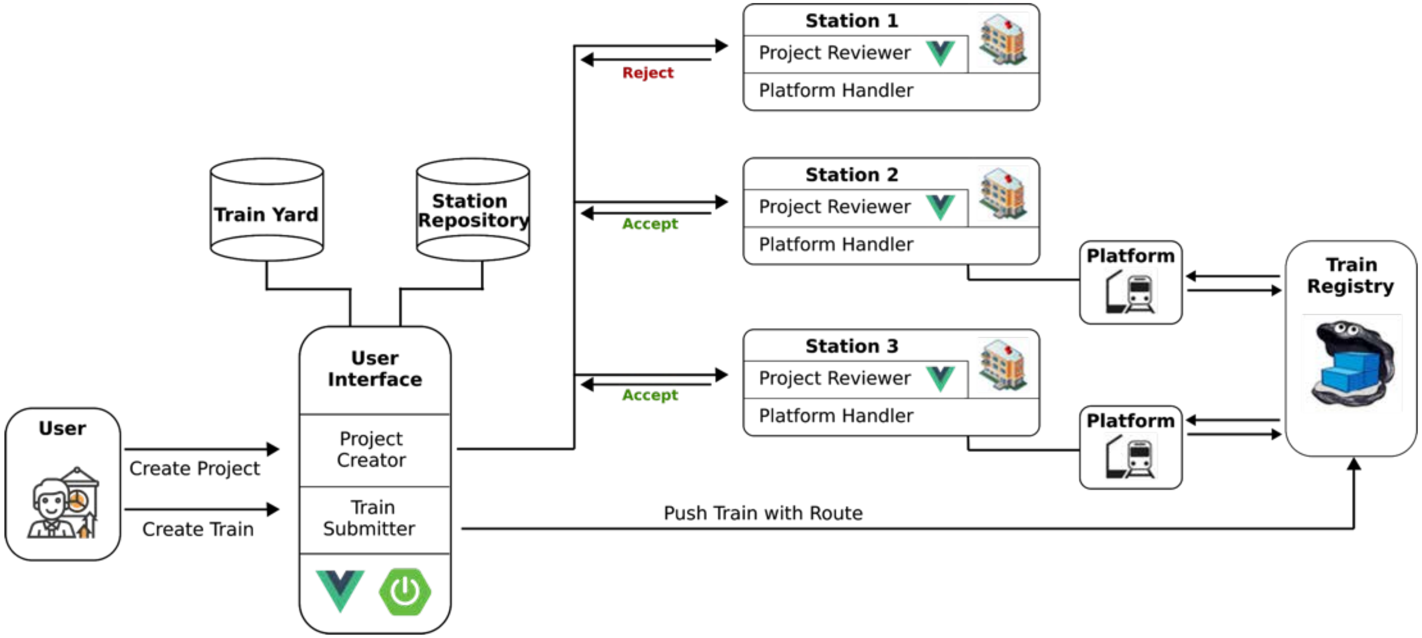
- Align concepts and reuse different components after FAIR standards
- Develop an infrastructure across boarders
- Submit joint grant proposals

Inter Consortial Work:

- Aachen & Leipzig – meta data and patient data availability
- Tübingen – analysis and security

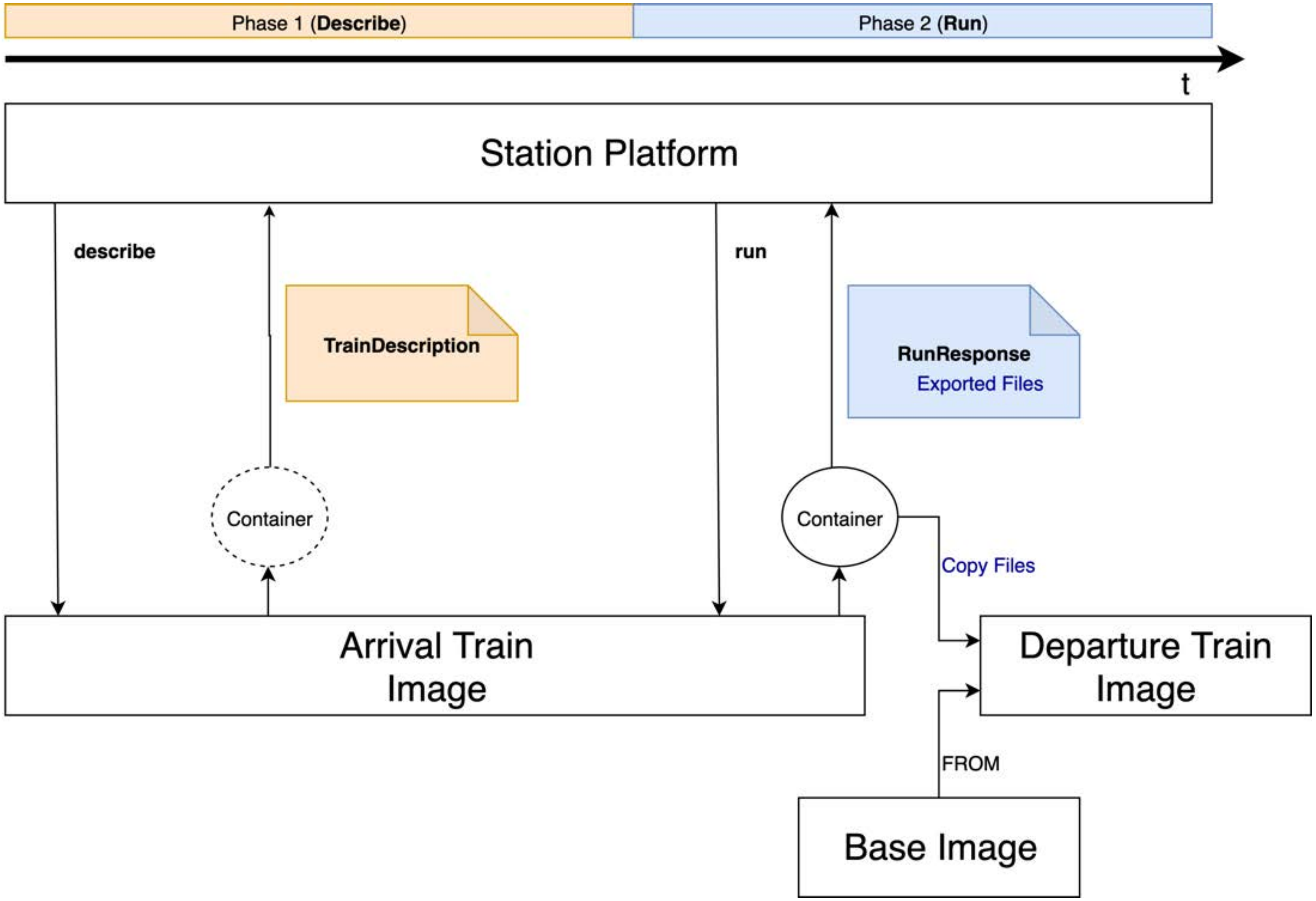
Previous work from Tübingen

- Overall workflow to submit and run trains



- Definition of trains based on docker
- How stations can execute trains
- Commands to communicate between train and stations

Previous execution of trains



Current Status

- Security concept to provide integrity, authenticity and confidentiality
- Deployed key management HashiCorp Vault
- Platform is in Python
- Extended Stations and Train-API with security
- Deployed registry to develop with realistic conditions

Current Status continued

- More staff and partners involved within PHT
- Extending and strengthening collaborations
- Proposals for several new use cases
- Code available at: https://gitlab.com/PersonalHealthTrain/implementations/germanmii/difuture
- Several concepts to extend architecture
- Currently focus on security and ML
  - Directly following: privacy and analysis

Limitations of PHT

Methods

- Currently only Online Machine Learning
- No non-linear SVMs
- No parallel execution and training of models

➢ Concepts to extend PHT to Secure Multi Party Computation and Federated Machine Learning

Security

– No detection of manipulation of trains

– Models are distributed unencrypted

➢ Secure the PHT

Start a secure train

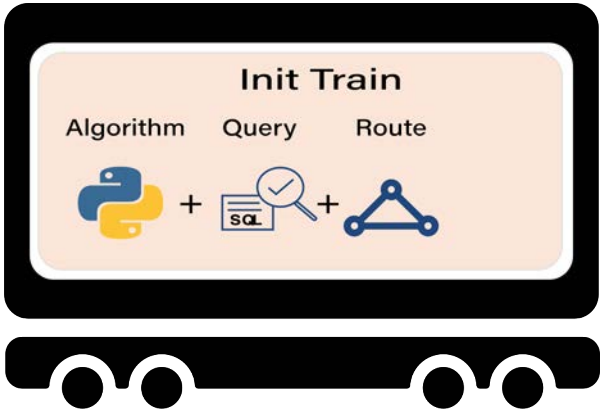
User (has public and private key) logs into central UI

User defines algorithm, query and contained stations

Step **1** / 4

- Train Builder (TB) matches PIDs to IDs of stations
- Creates route

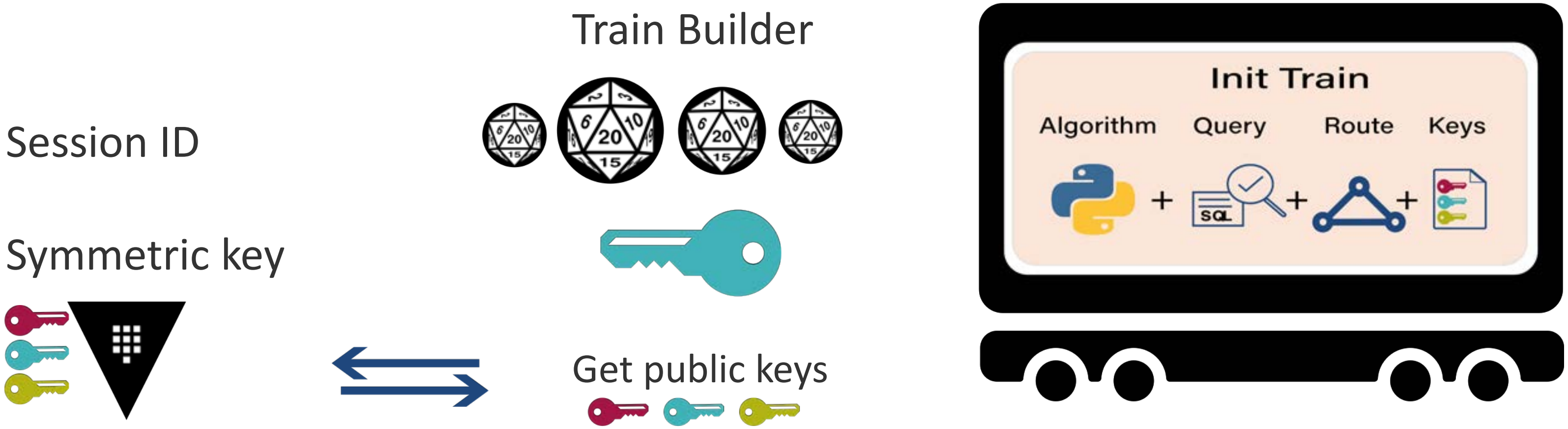
Station ID	User		Train Builder
EKUT	1	→	Xzgf7a
UKA	0		Zfq4az
MRI	1	→	4dgaRi
KUM	1	→	PdFa2a



Start a secure train

Step **2** / 4

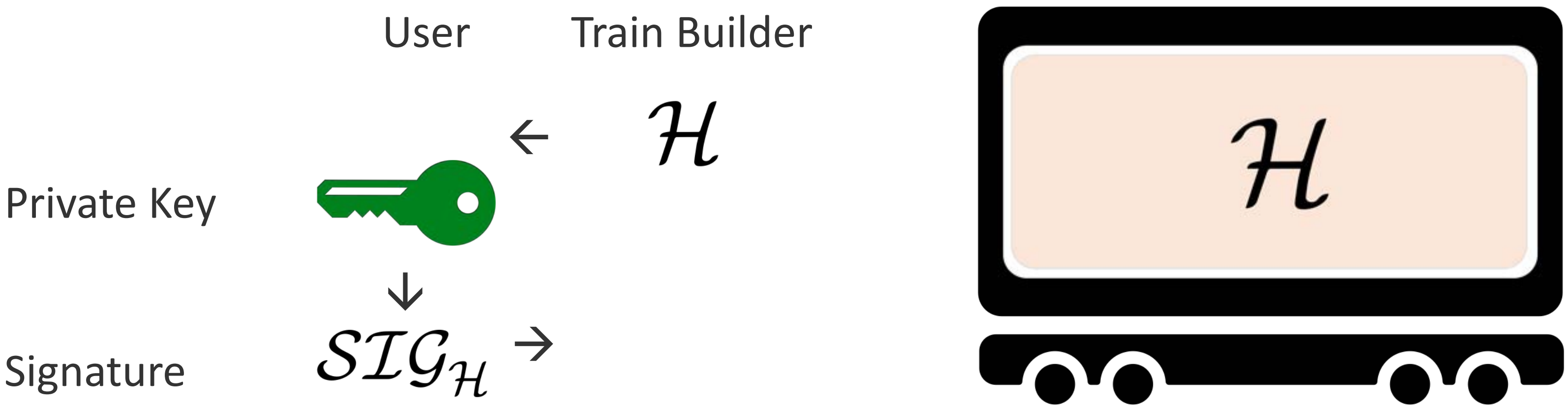
- TB creates session ID and symmetric keys
- TB receives all public keys of participating stations



Start a secure train

Step **3** / 4

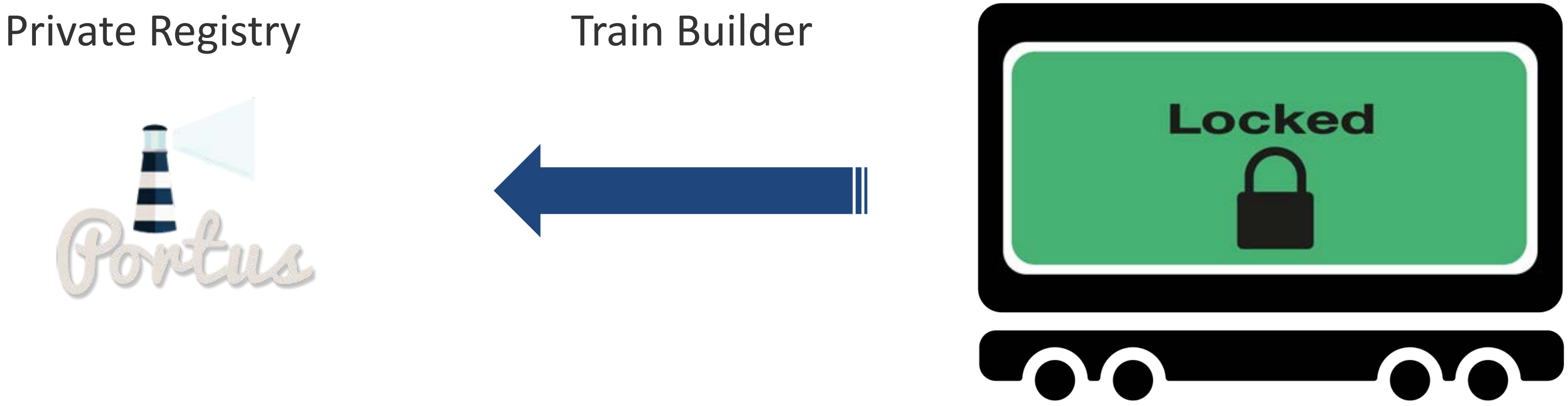
- TB calculates hash of files
- User signs with his private key



Start a secure train

Step 4 / 4

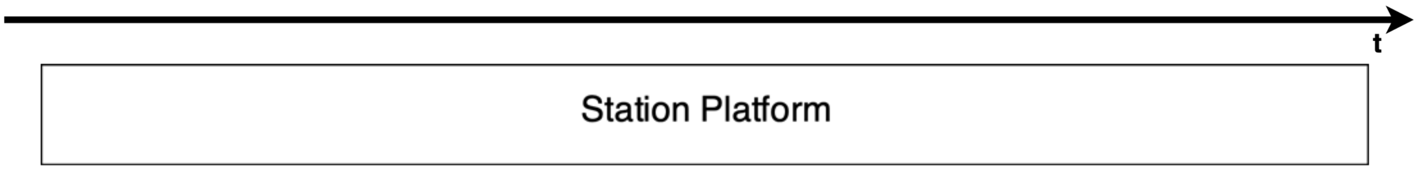
- TB locks train
- TB pushes train to private docker registry



Private Registry

Train Builder

Secure execution of trains

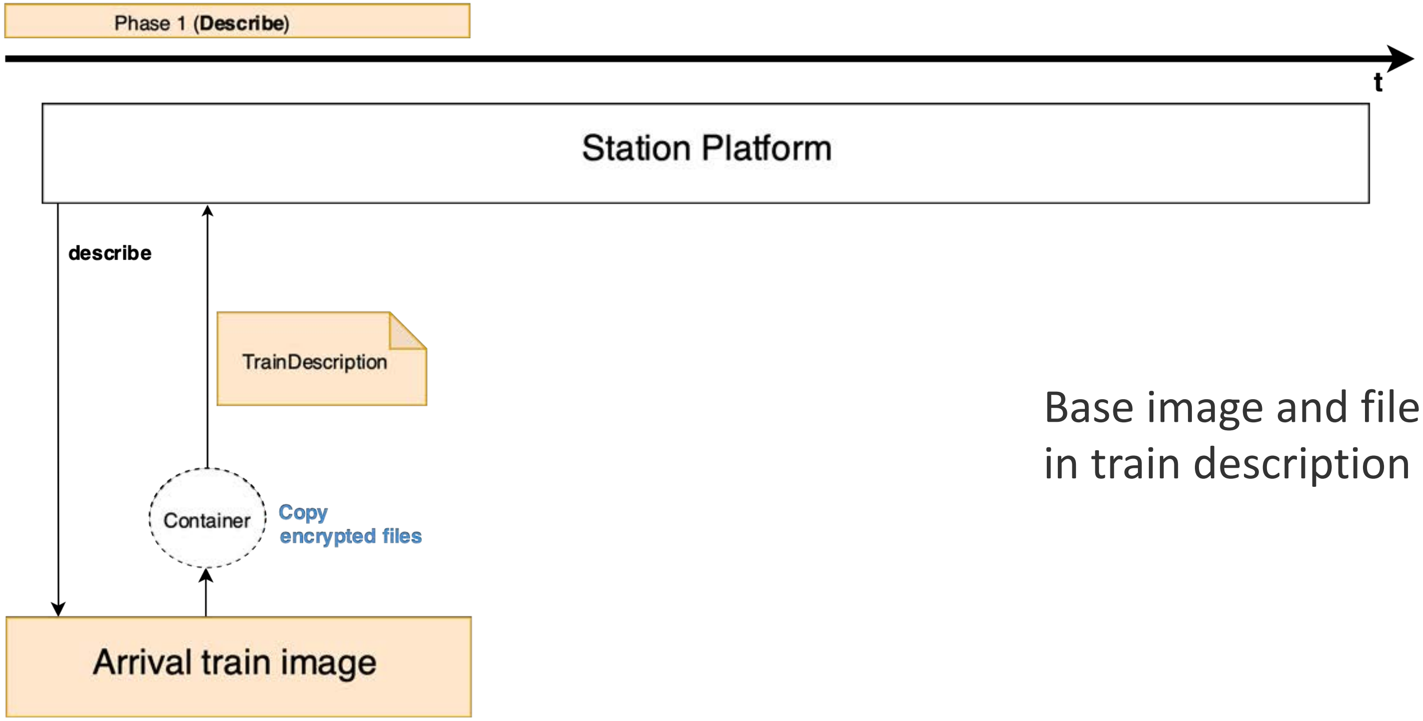


Each station needs:

Public key

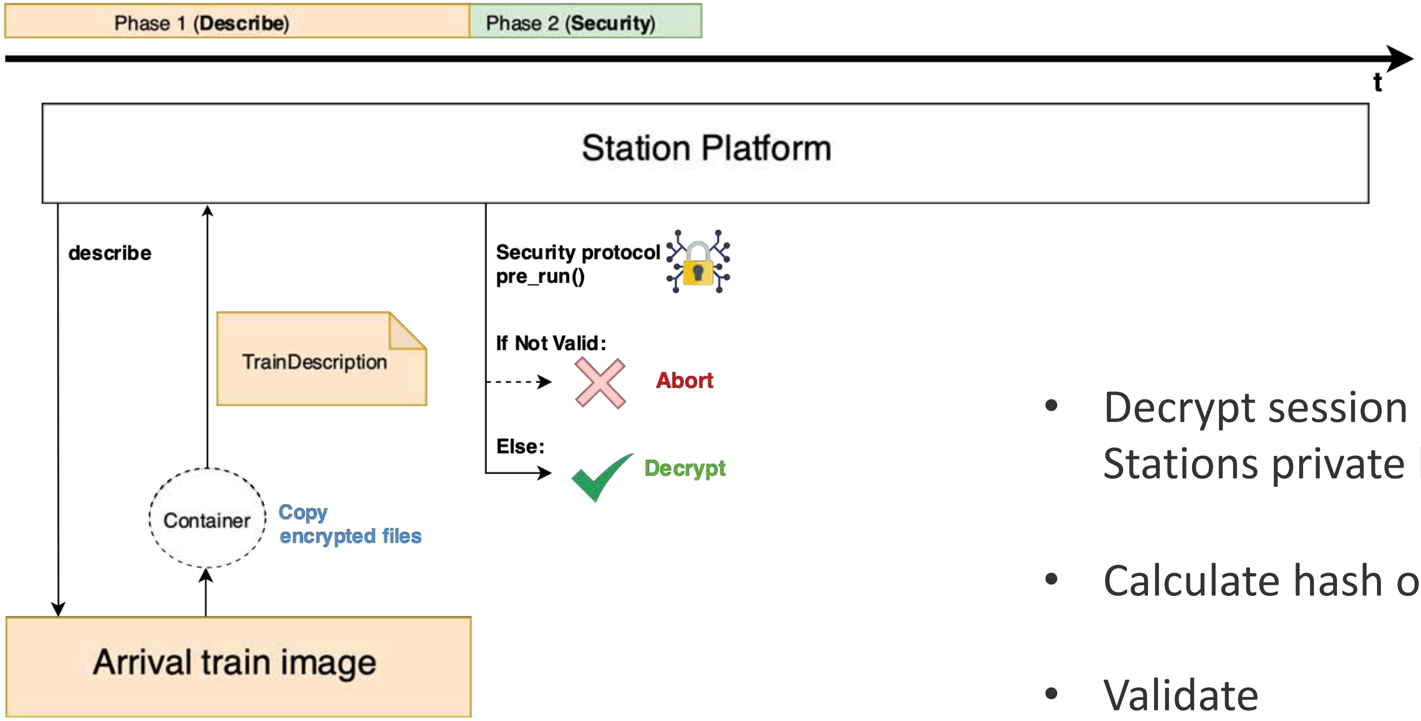
Private key

Secure execution of trains



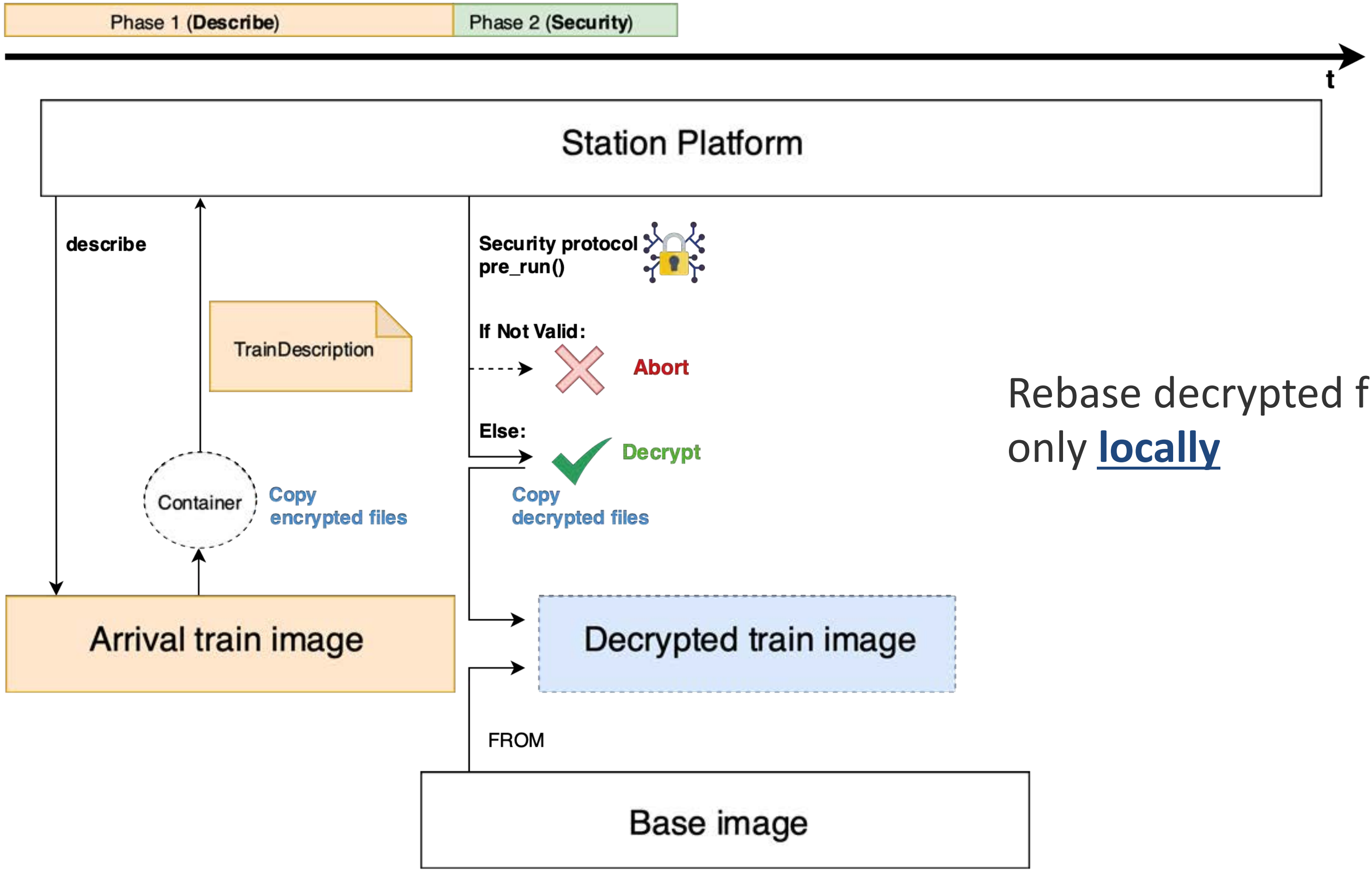
Base image and files are specified in train description

Secure execution of trains



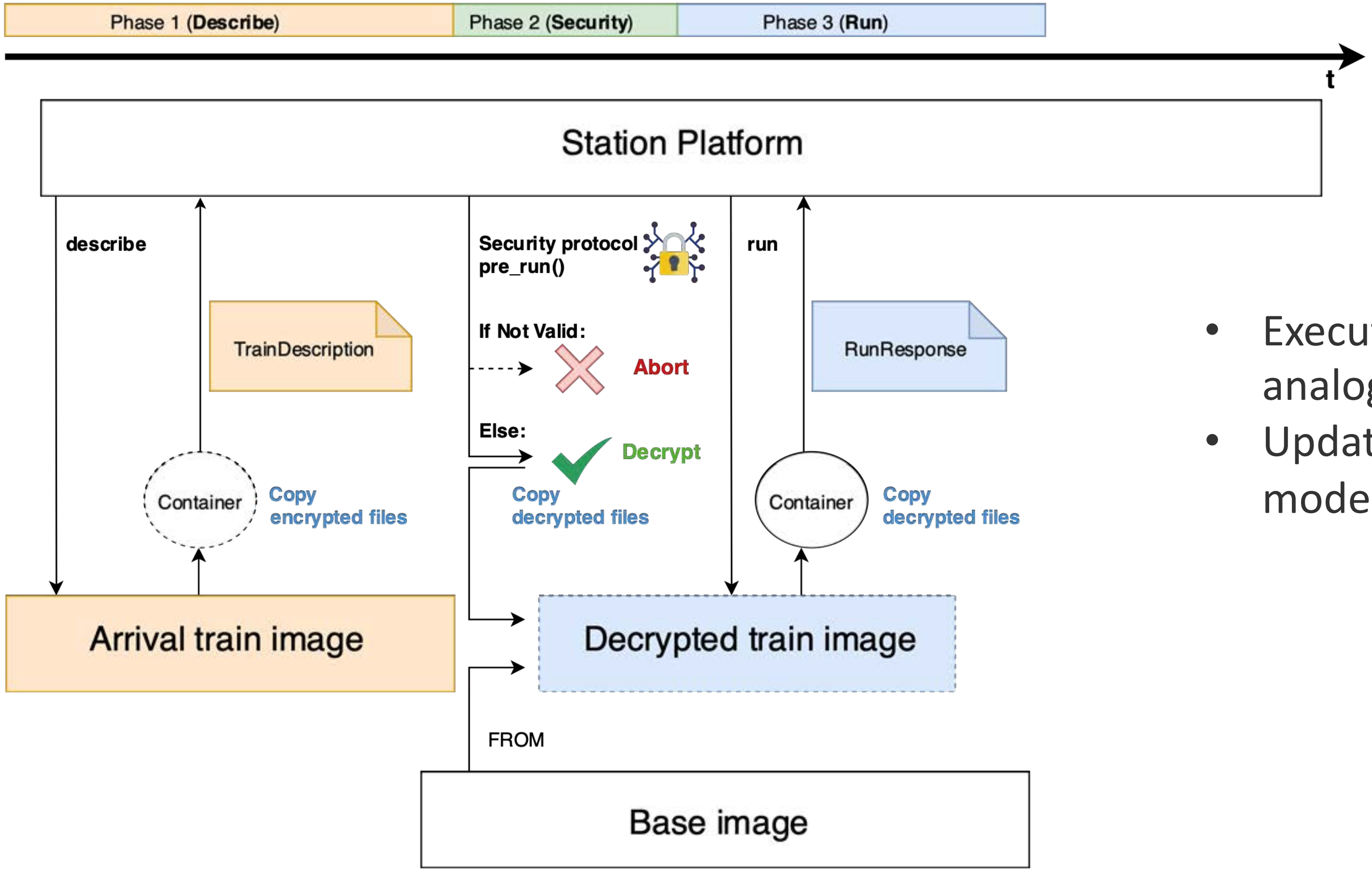
- Decrypt session key with Stations private key
- Calculate hash of files
- Validate

Secure execution of trains



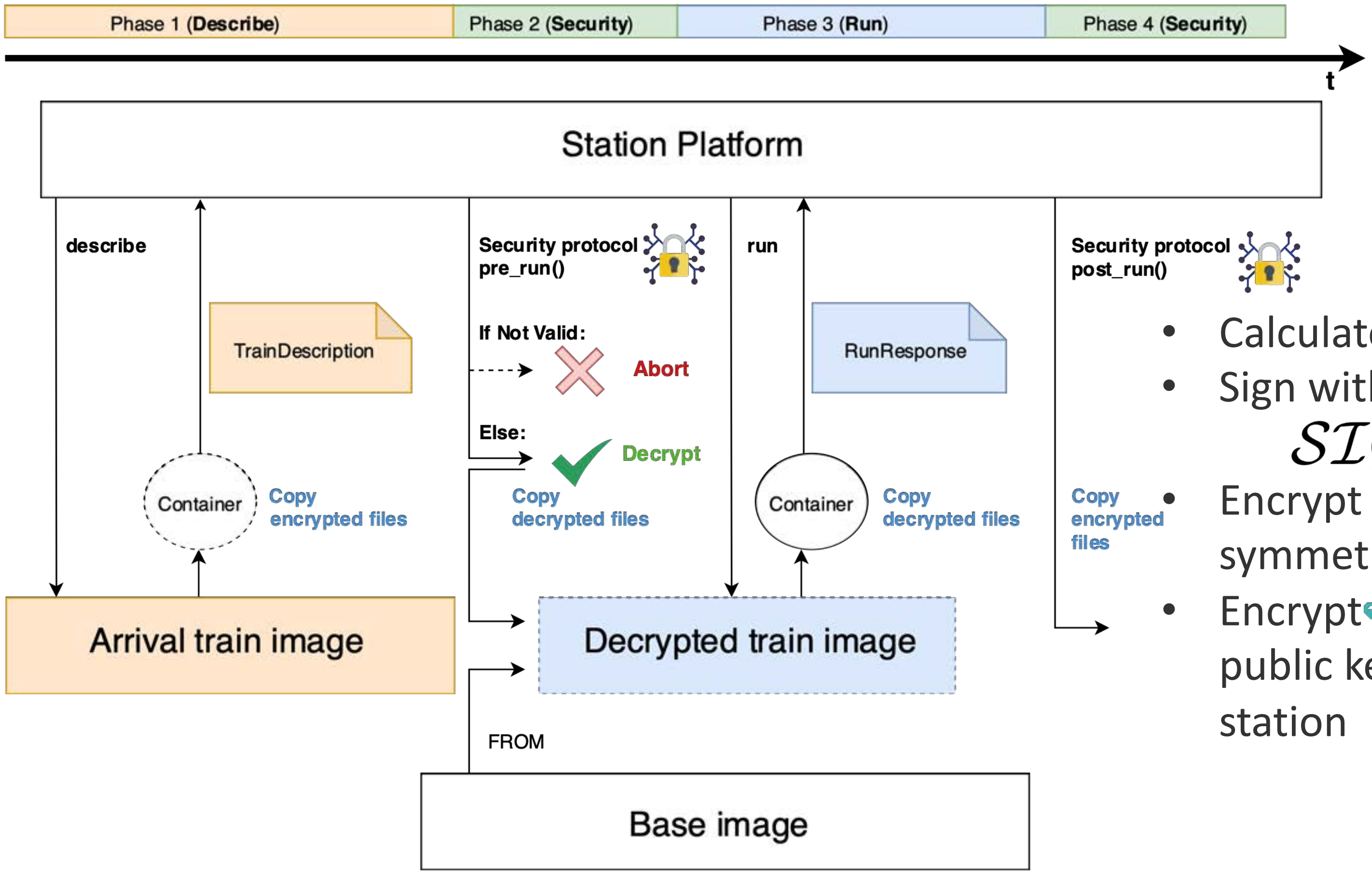
Rebase decrypted files to execute only **locally**

Secure execution of trains



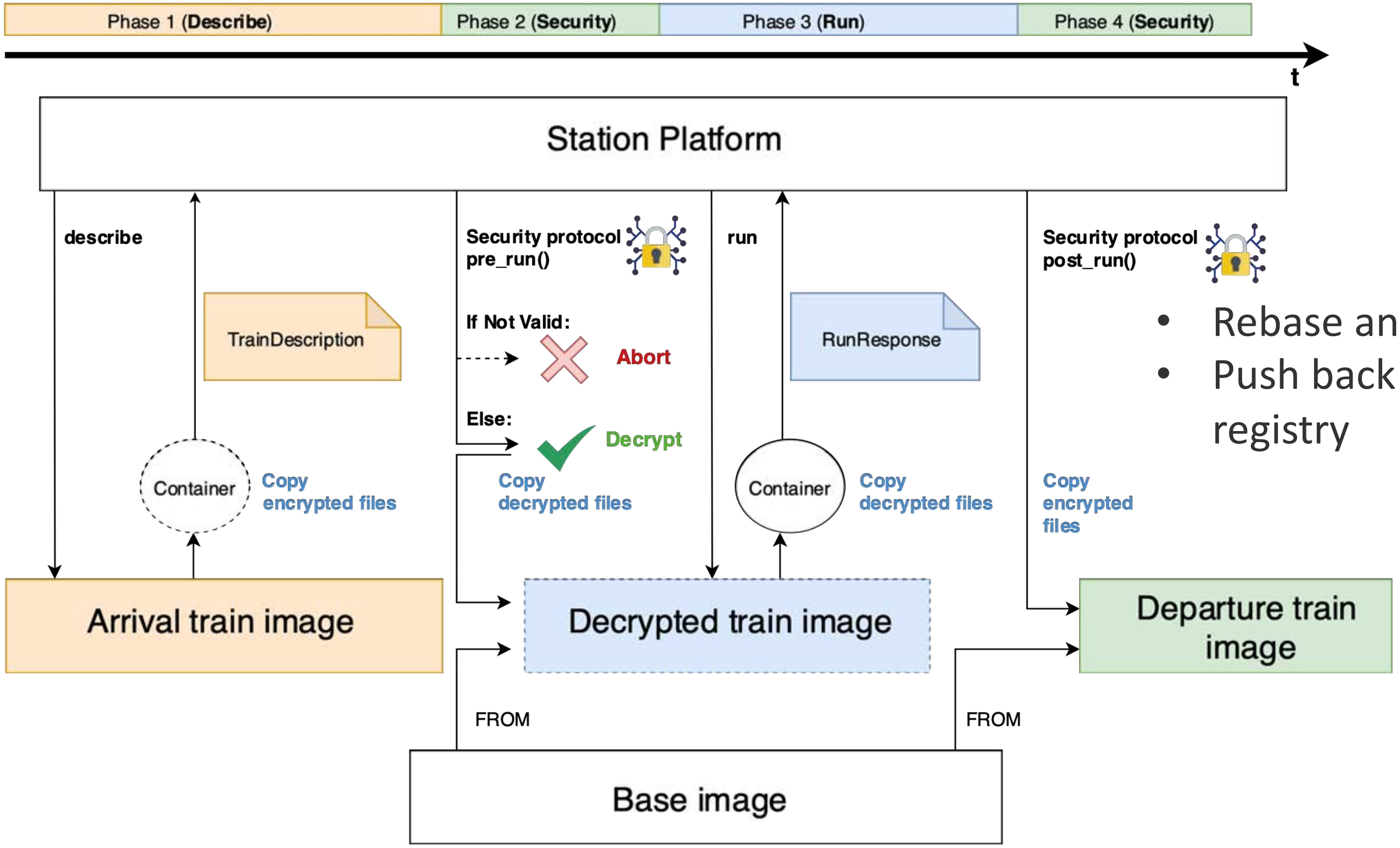
- Execute train analogue
- Update previous model

Secure execution of trains



- Calculate hash $\mathcal{H}$
- Sign with private key $\mathcal{SIG}_{\mathcal{H}}$
- Encrypt files with symmetric 🔑
- Encrypt 🔑 only with public key of next station

Secure execution of trains



- Rebase and tag
- Push back to registry

Performance Test Trains

- Tested with four different trains
- Each train has the same task:
  - Distribute model with fixed size
- Size of model to encrypt differs

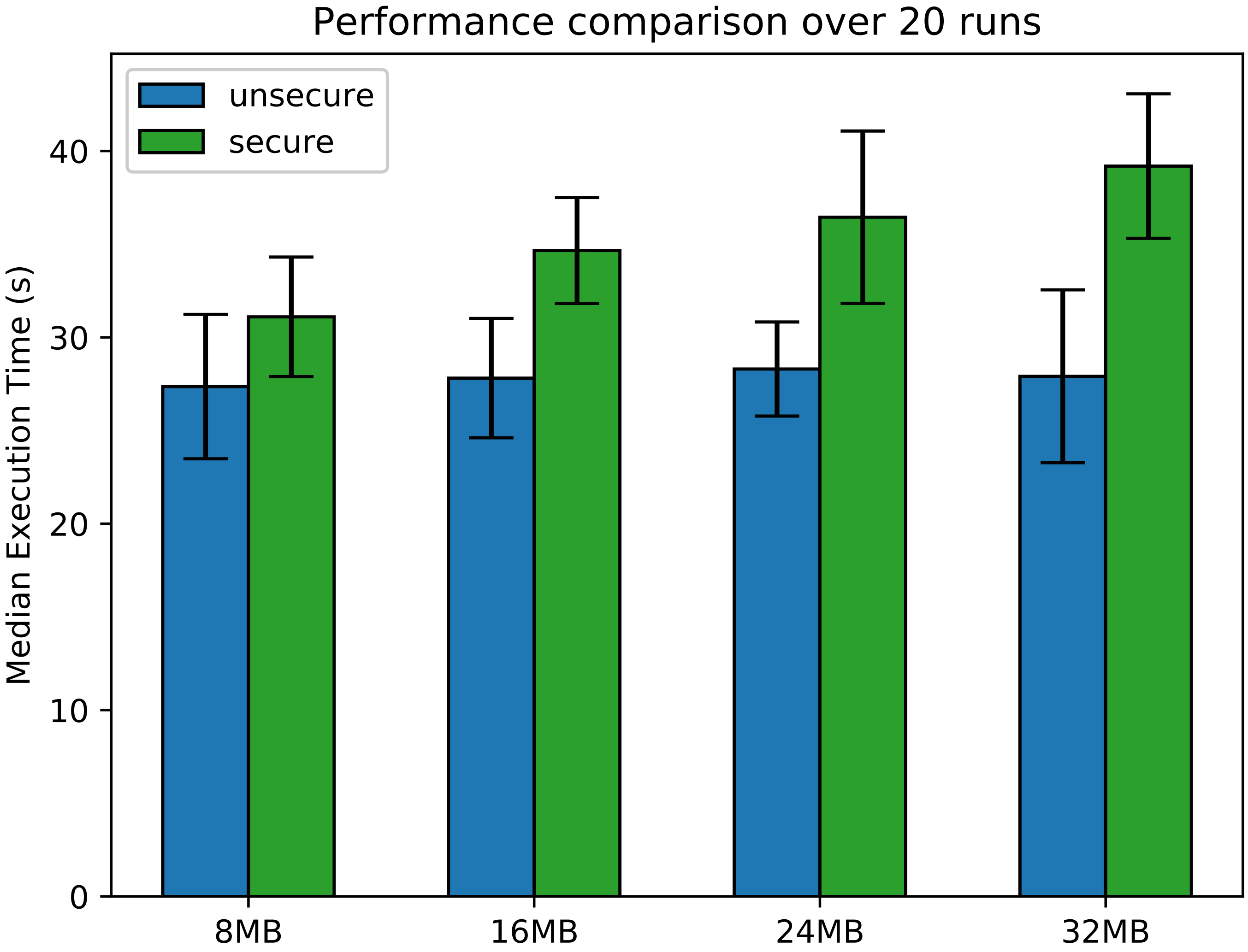
Train	Matrix size	Size (MB)
1	1024 x 1024	8
2	2056 x 1024	16
3	2056 x 1536	24
4	2056 x 2056	32

Table 1: Train model sizes

Component	Specification
CPU	Intel® Core™ i5-8400
RAM	16 GB DDR4
OS	Ubuntu 18.04

Table 2: Hardware specs of test platform

Results


Performance comparison over 20 runs
unsecure
secure
Median Execution Time (s)
0
10
20
30
40
8MB
16MB
24MB
32MB

Security Conclusion

- Only stations and submitting user have access to the final analysis results
- An adversary compromising the registry can neither impersonate stations nor users nor access or change analysis results
- Stations do not share secrets with the user, Docker Registry and other stations
- Execution time increases linearly with model size

Outlook

- Merge national PHT architectures
  - Overall agreement on PHT workflow ✓
  - Specify interfaces between central services 🚧
- Participant-level differential privacy against inference attacks on ML models
- Extend current PHT architecture to enable additional methods required from use cases
- Define a governance to use the PHT

DIFUTURE Team Tübingen

Special Thanks:
Mete Akgün,
Felix Bötte,
Michael Graf,
Oliver Kohlbacher,
Florian König,
Jörg Peter,
Nico Pfeifer,
Sascha Rehm,
Felix Sieghörter,
Lukas Zimmermann

Acknowledgements

SMITH Smart Medical Information Technology for Healthcare

miracum Medical Informatics in Research and Care in University Medicine

TECHNISCHE UNIVERSITÄT MÜNCHEN

LMU

Universität Augsburg University

UNI FREIBURG

DIFUTURE Data Integration for Future Medicine

GO FAIR

RWTH AACHEN UNIVERSITY

LUMC

UM

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung