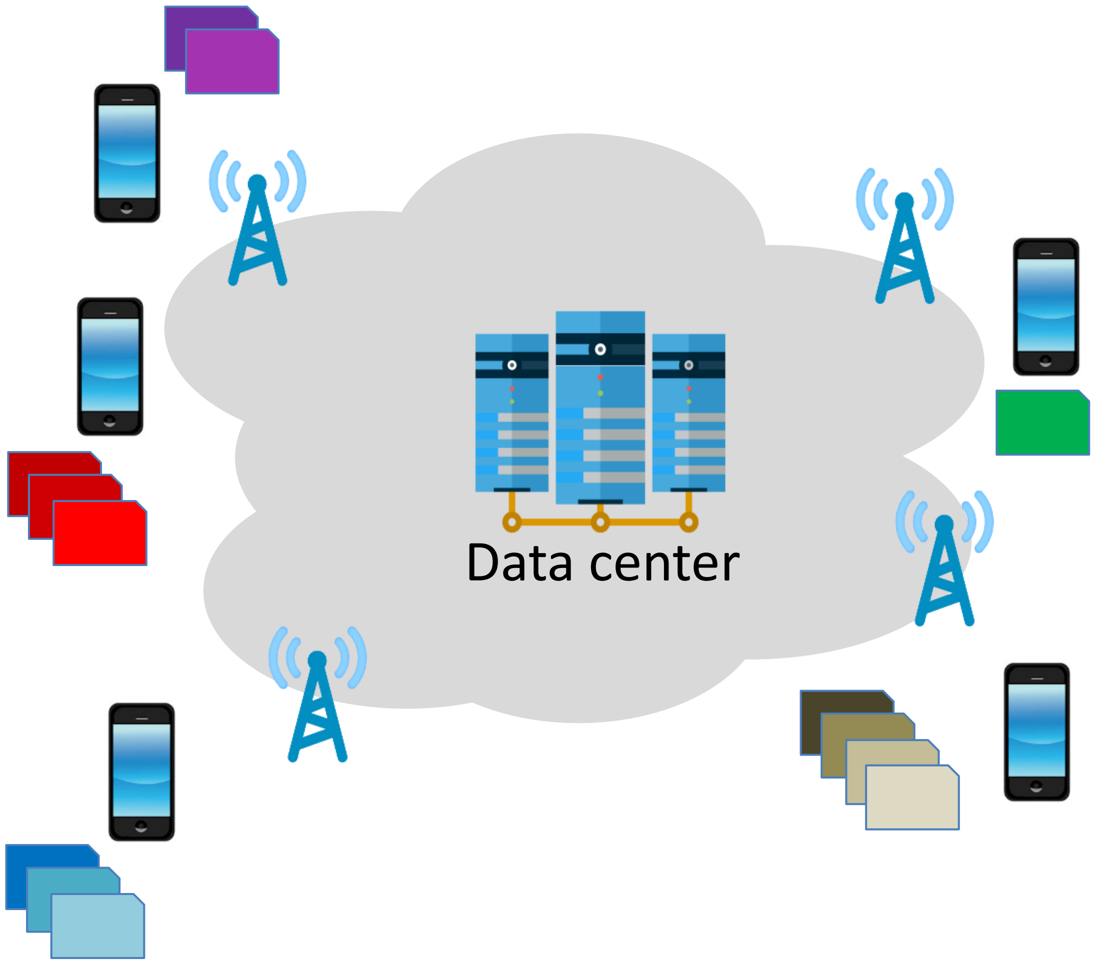
Federated Learning for Data Streams

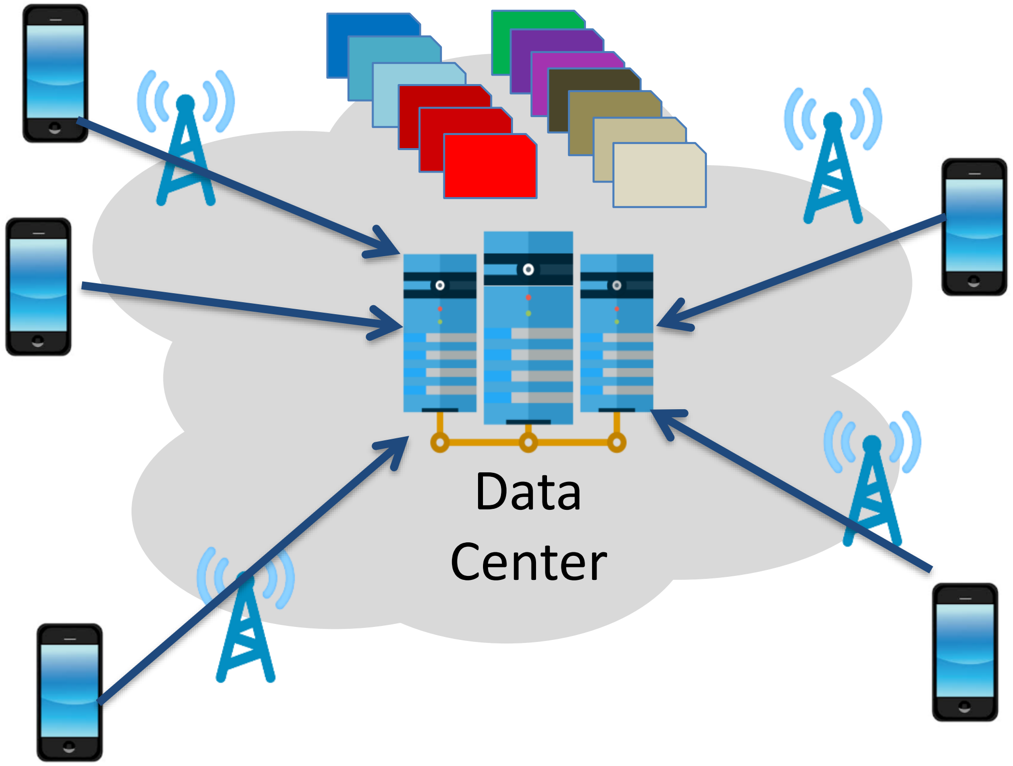
Othmane Marfoq[1, 2]   Giovanni Neglia[1]   Richard Vidal[2]   Laetitia Kameni[2]

[1]Inria, Université Côte d'Azur and [2]Accenture Labs

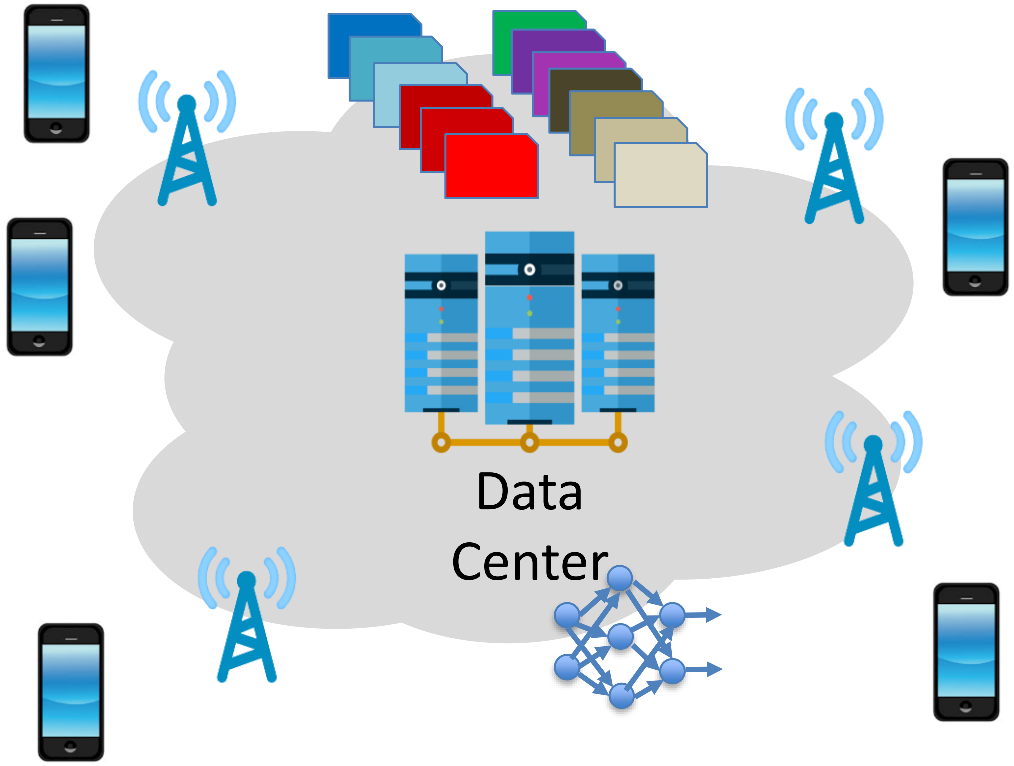
From Cloud Machine Learning to Federated Learning



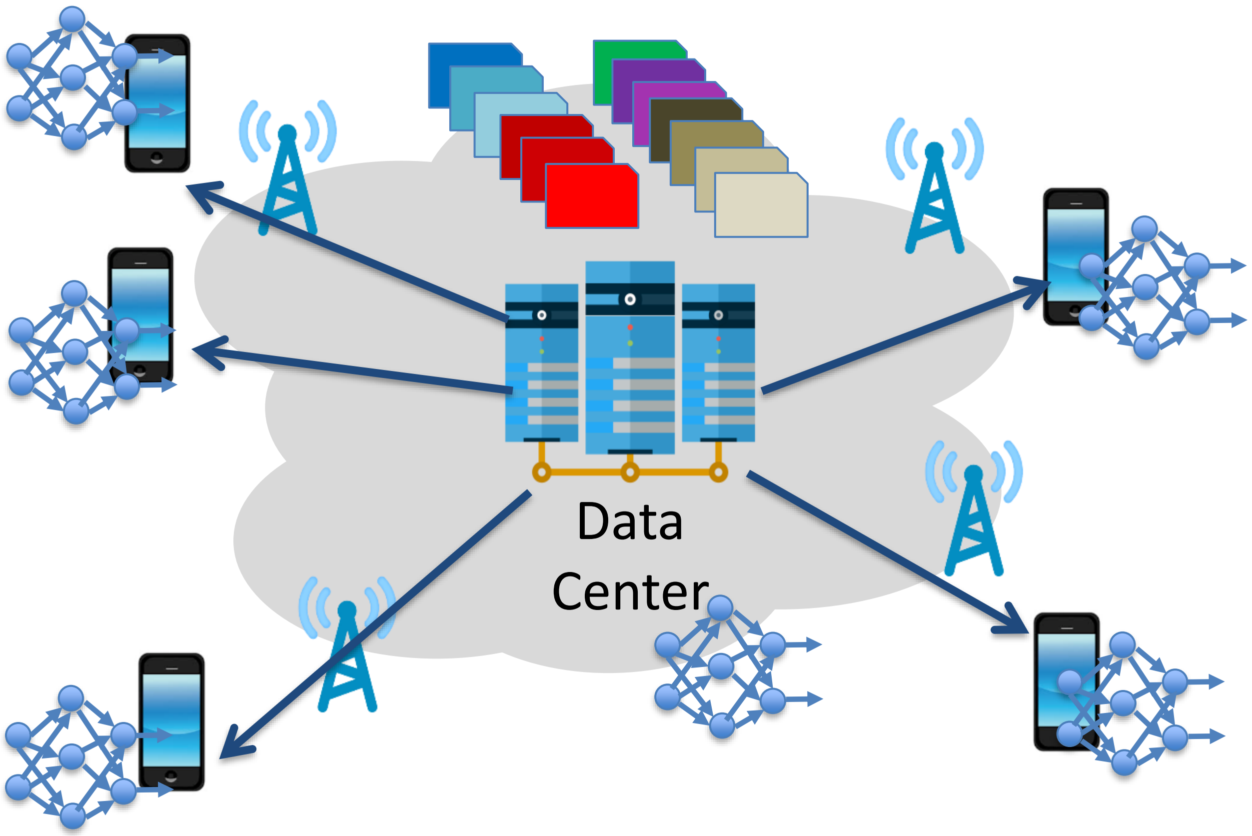
From Cloud Machine Learning to Federated Learning



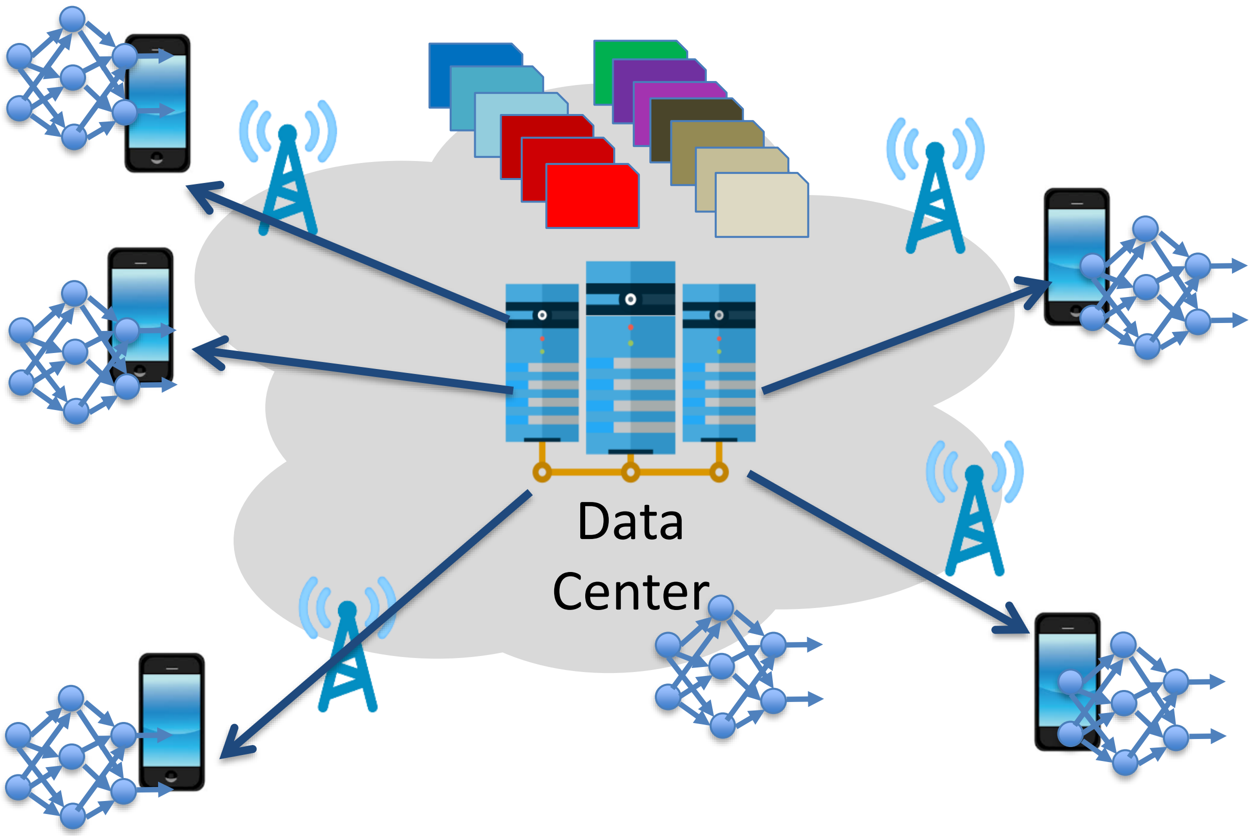
From Cloud Machine Learning to Federated Learning



From Cloud Machine Learning to Federated Learning



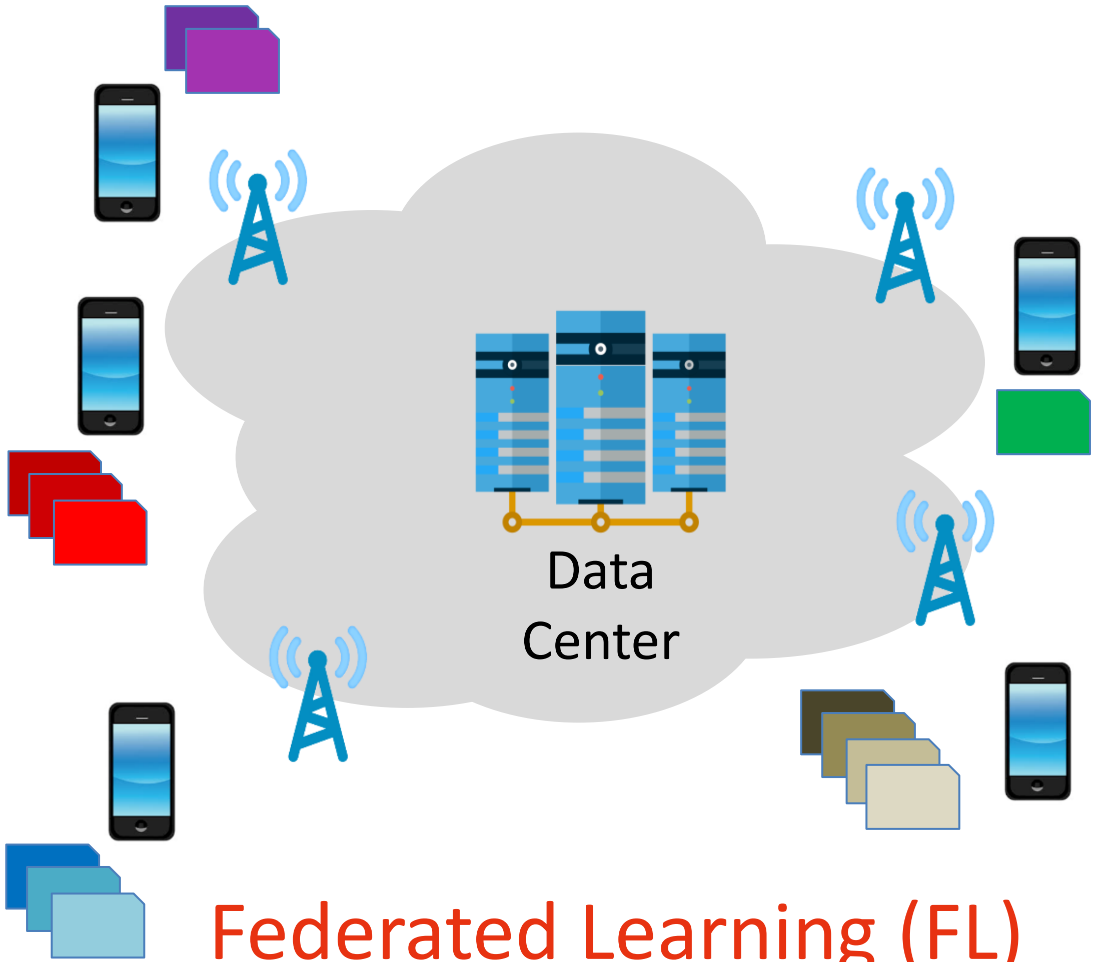
From Cloud Machine Learning to Federated Learning



Limitations:

1. communication cost
2. privacy concerns

From Cloud Machine Learning to Federated Learning

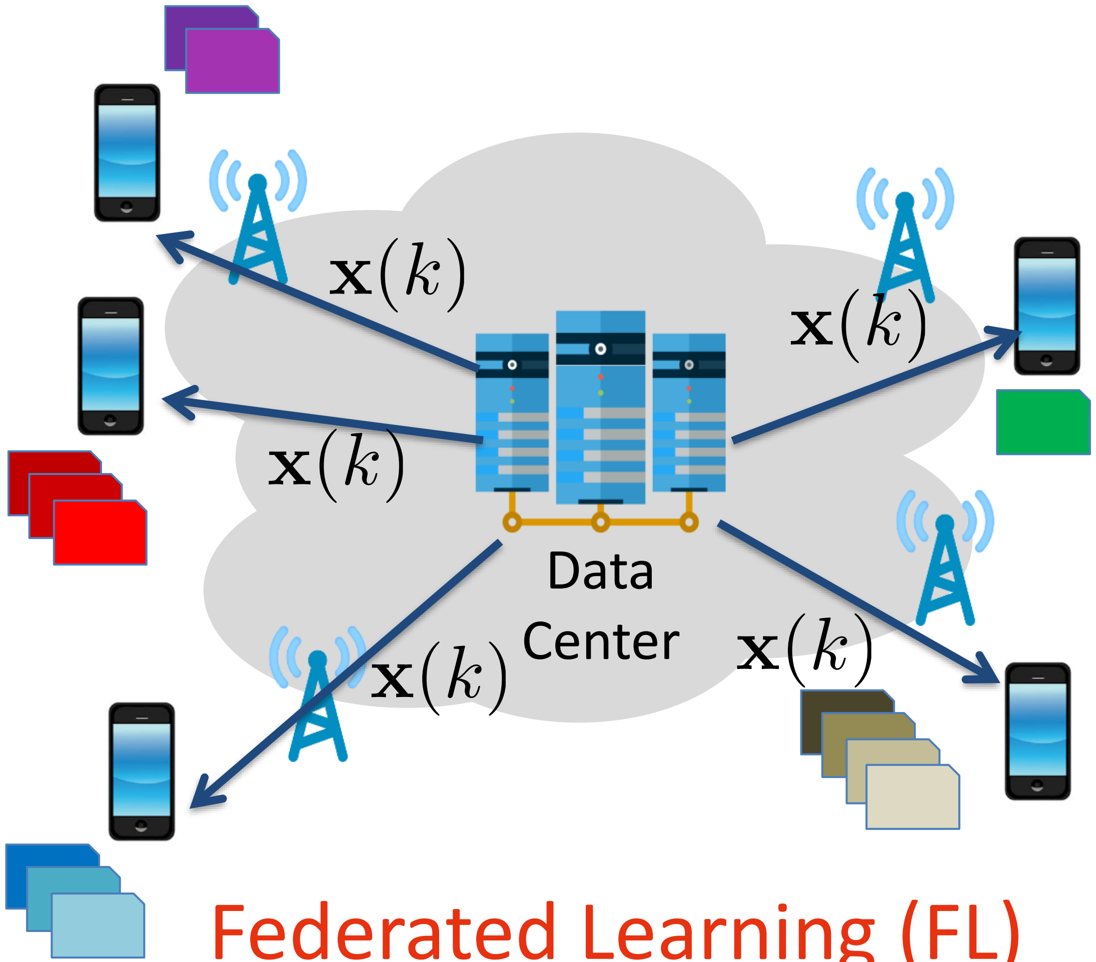


Federated Learning (FL)

Solution:

- keep clients' data on device
- only exchange model's parameters

From Cloud Machine Learning to Federated Learning

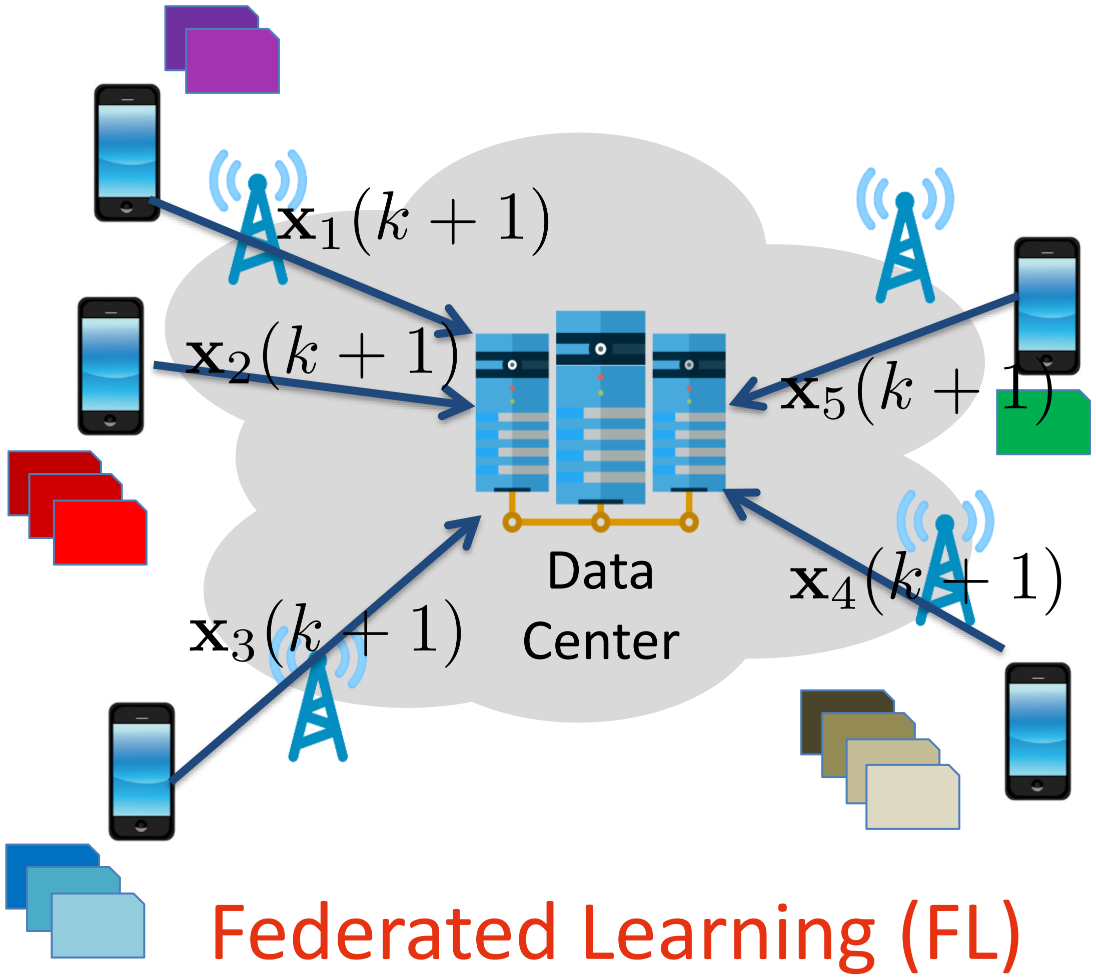


Federated Learning (FL)

Solution:

- keep clients' data on device
- only exchange model's parameters

From Cloud Machine Learning to Federated Learning

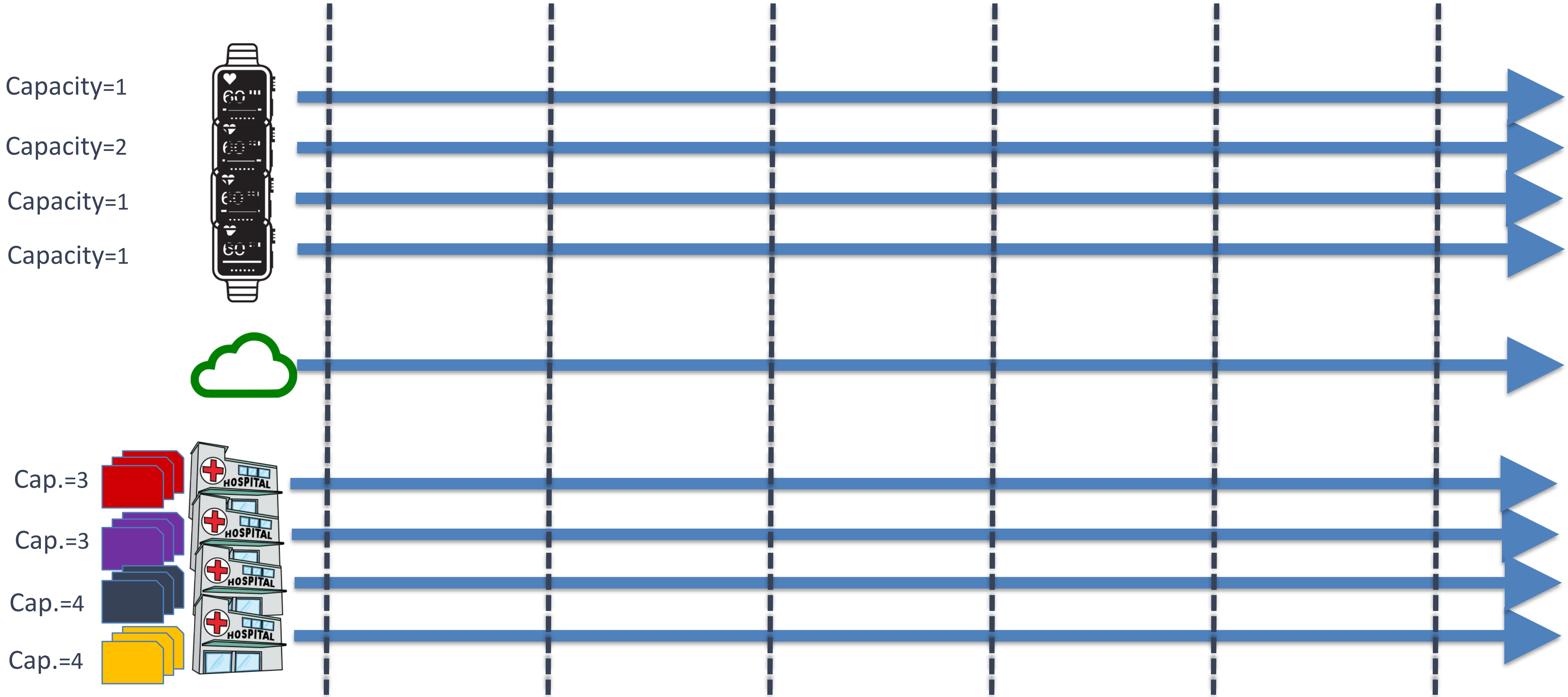


Federated Learning (FL)

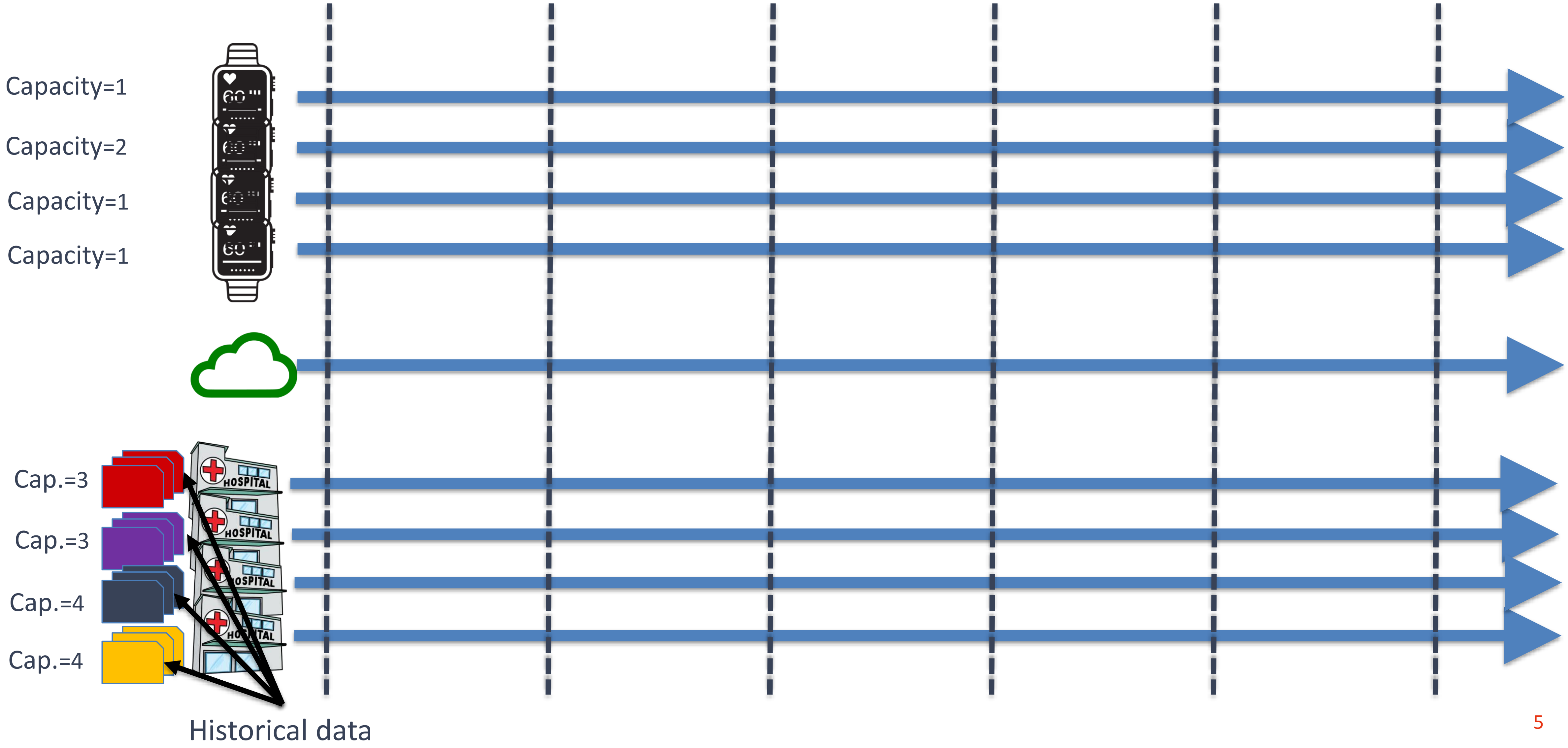
Solution:

- keep clients' data on device
- only exchange model's parameters

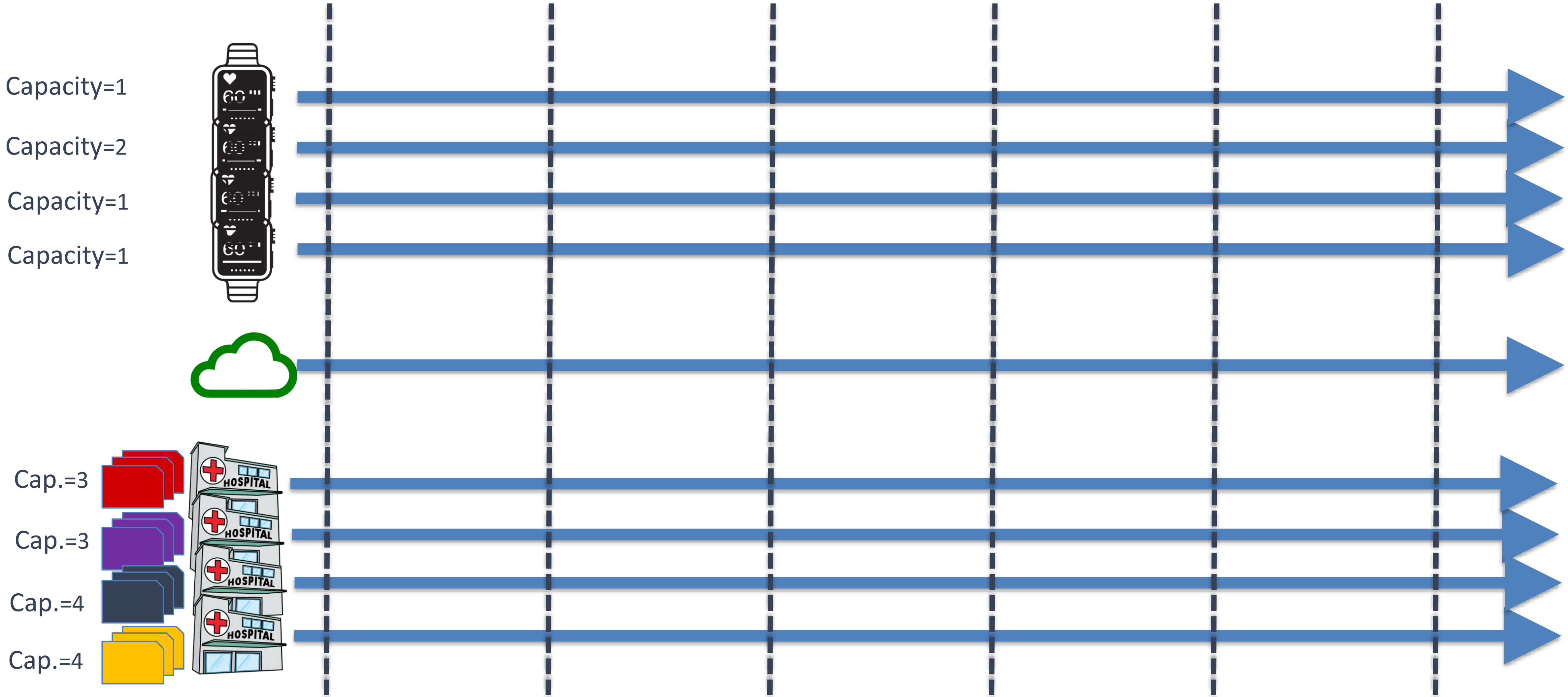
Federated Learning for Data Streams



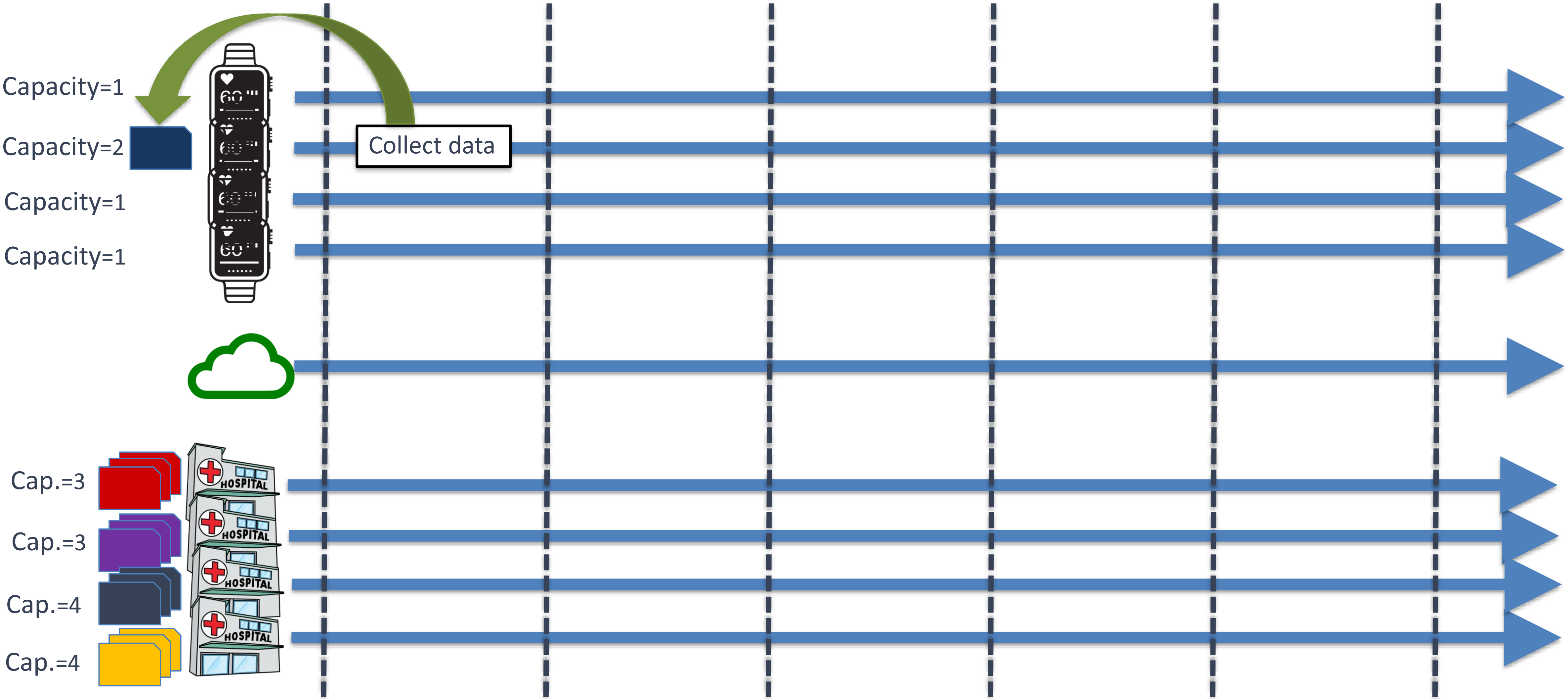
Federated Learning for Data Streams



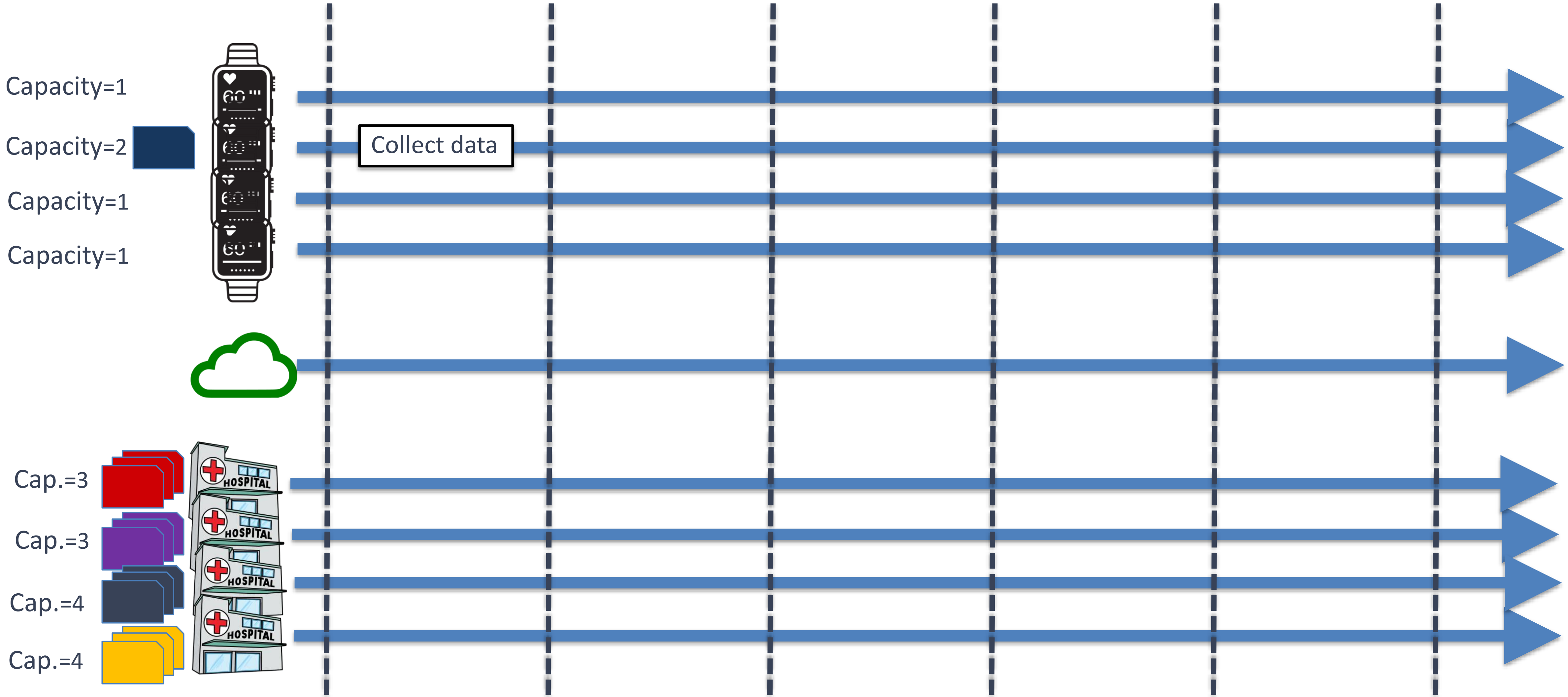
Federated Learning for Data Streams



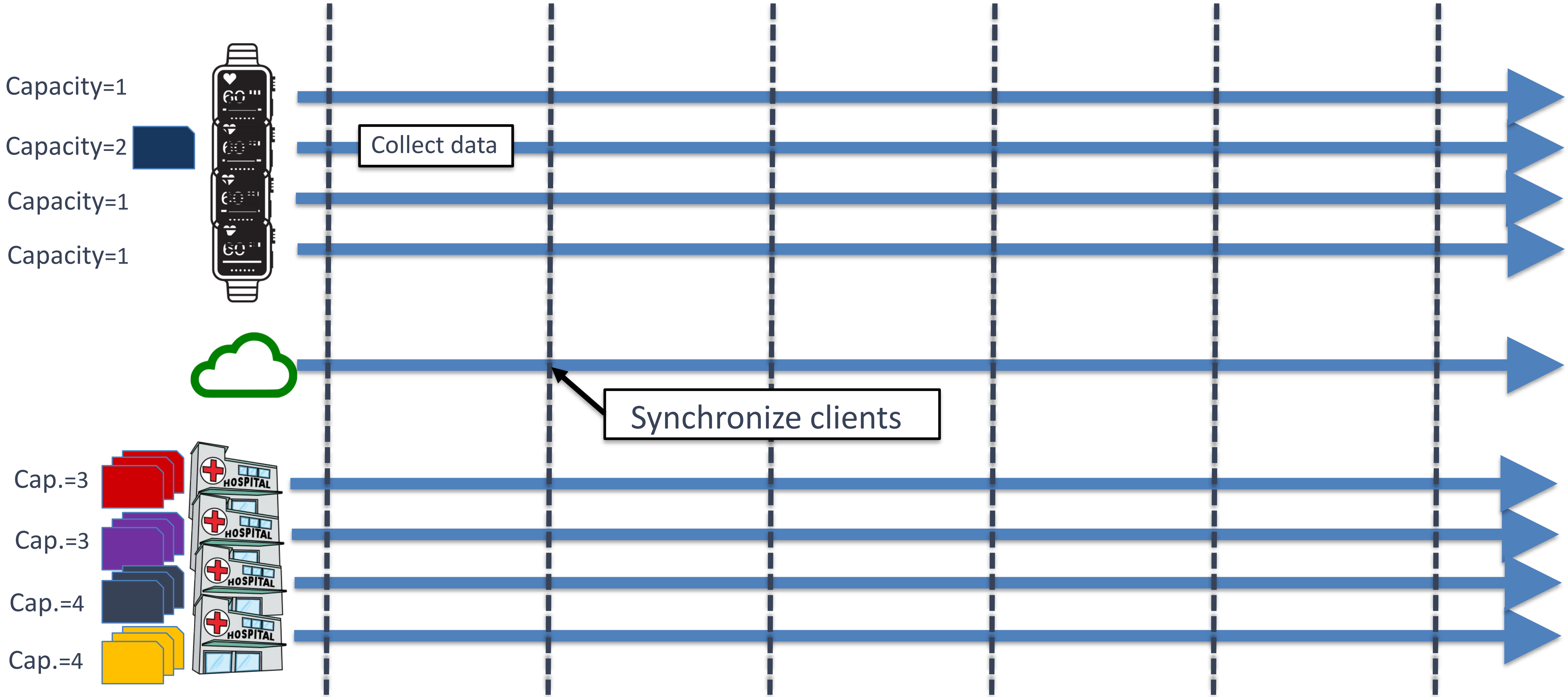
Federated Learning for Data Streams



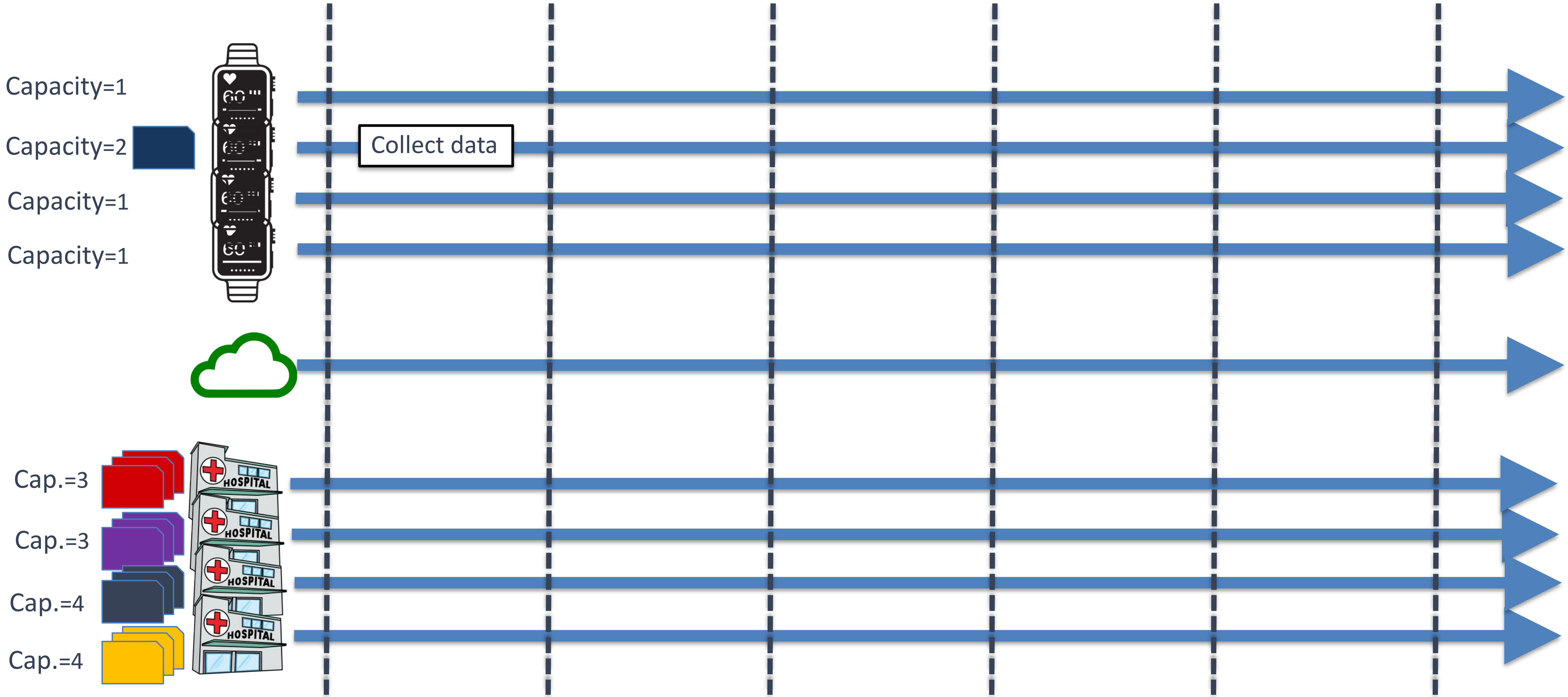
Federated Learning for Data Streams



Federated Learning for Data Streams

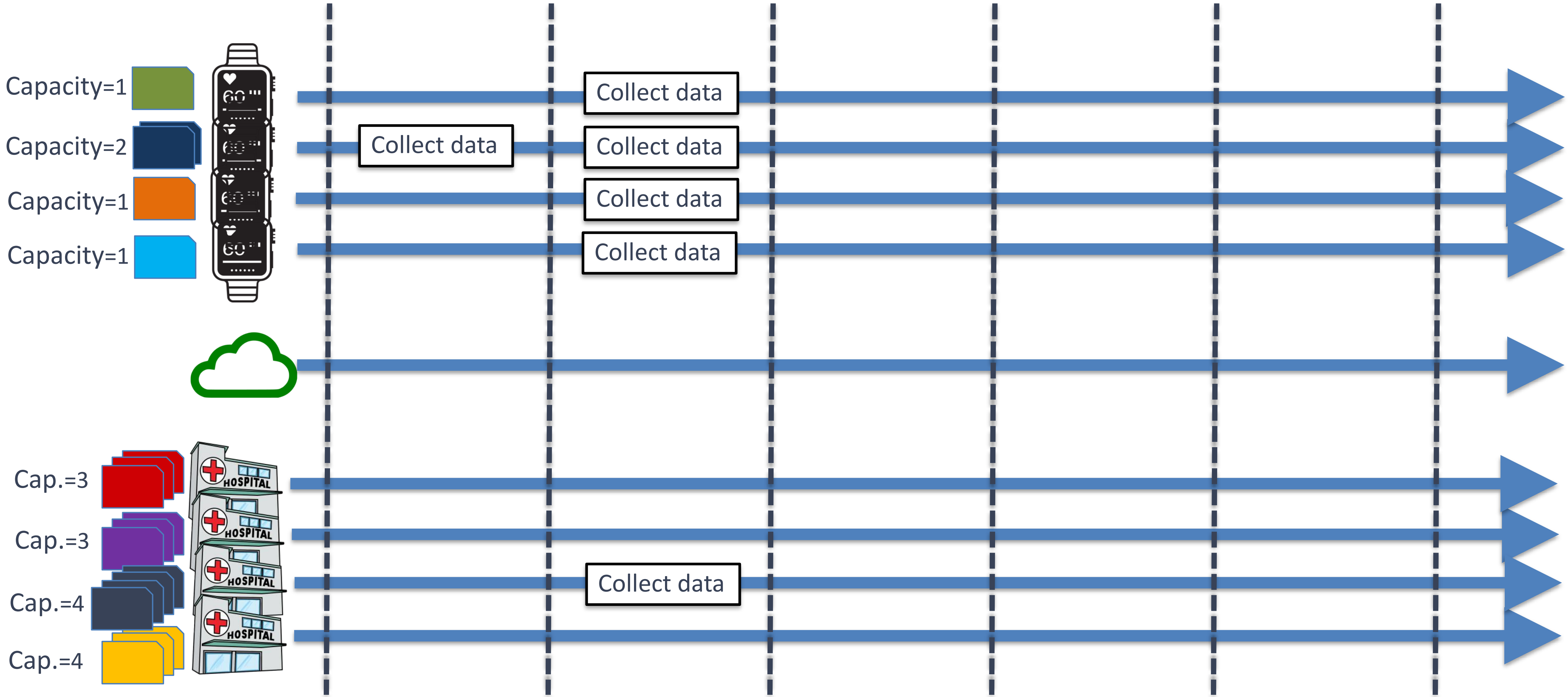


Federated Learning for Data Streams

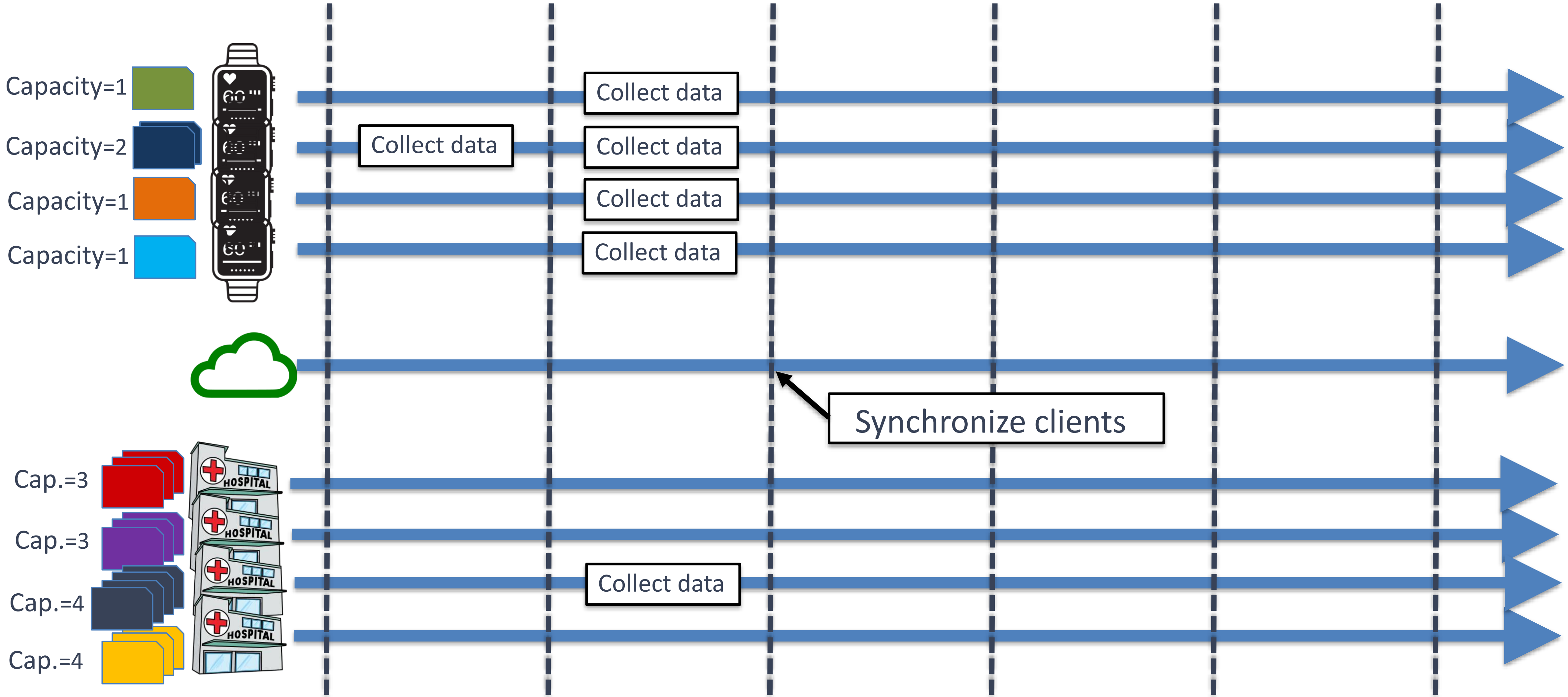


Federated Learning for Data Streams

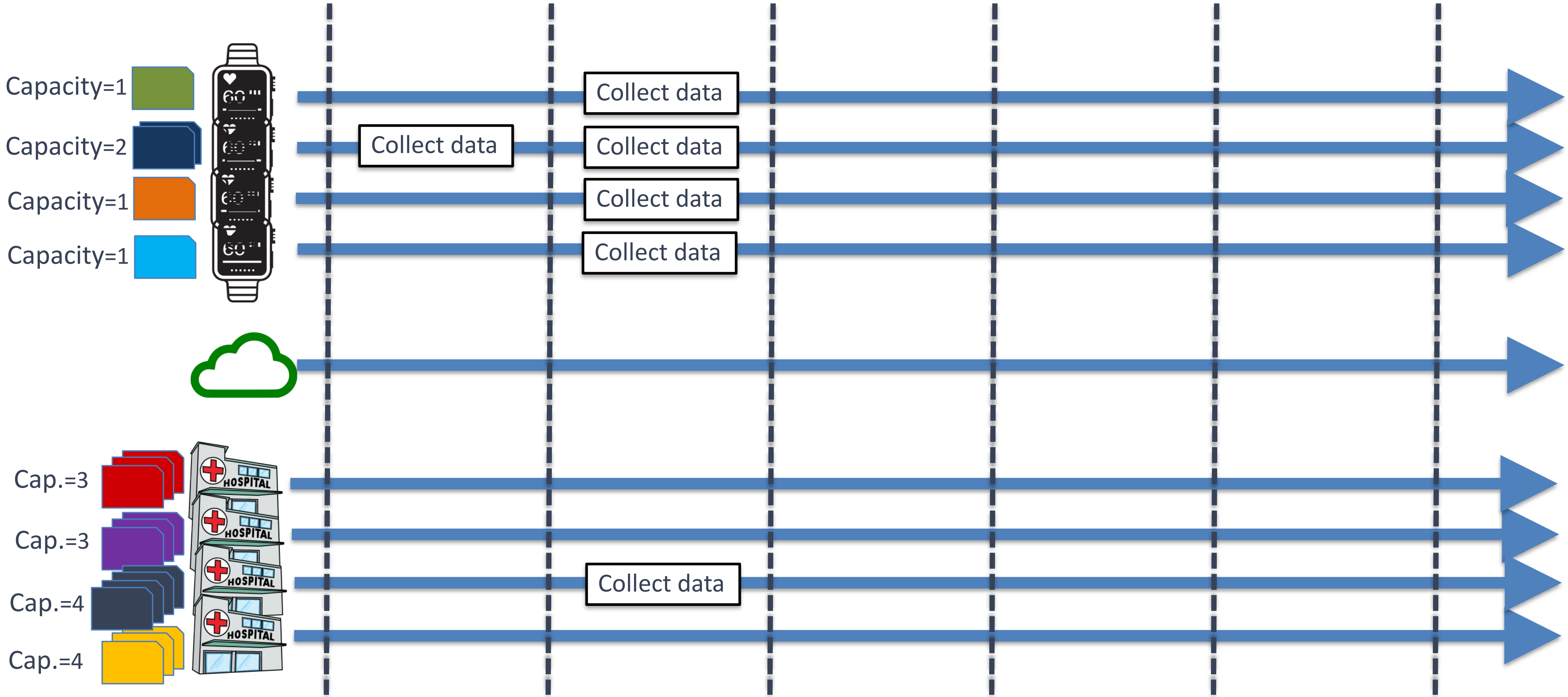
Federated Learning for Data Streams



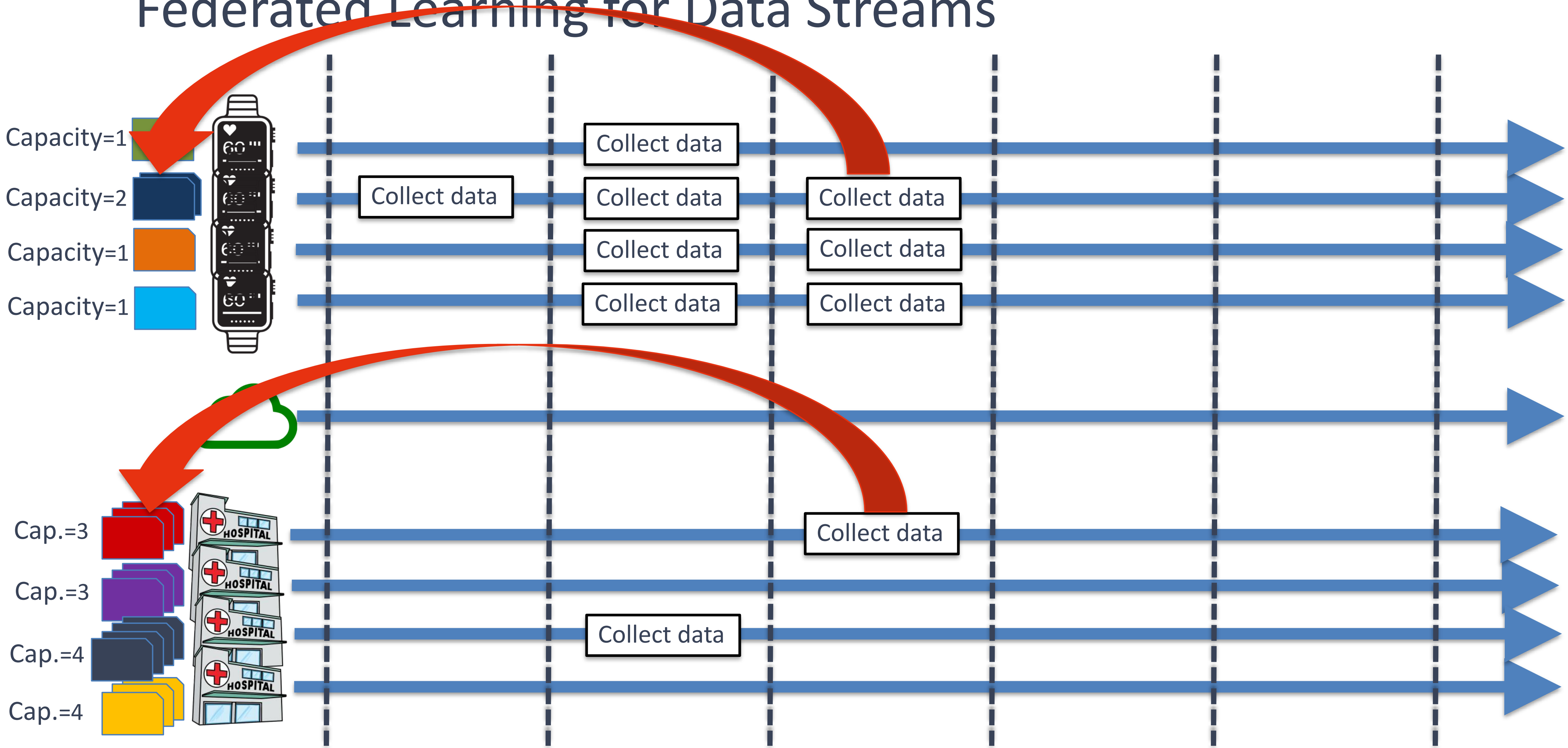
Federated Learning for Data Streams



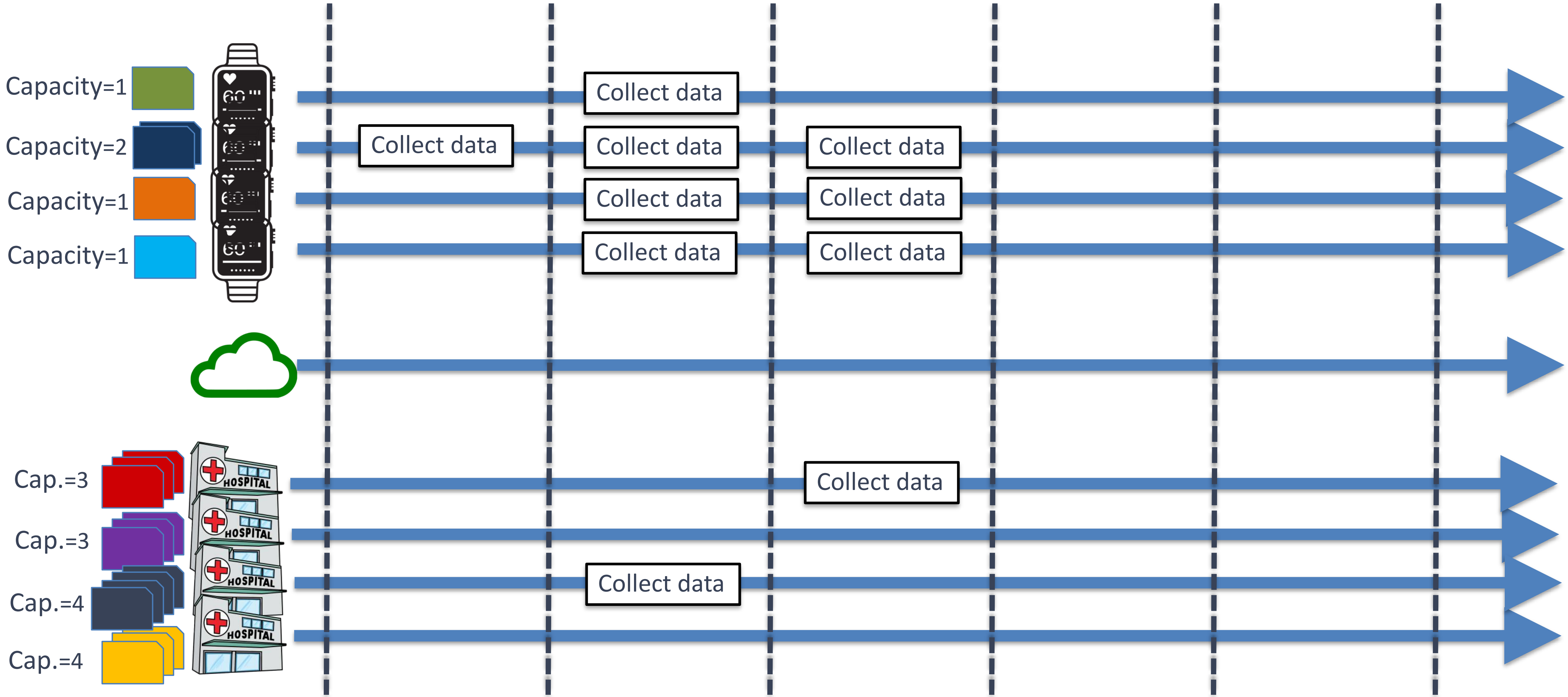
Federated Learning for Data Streams



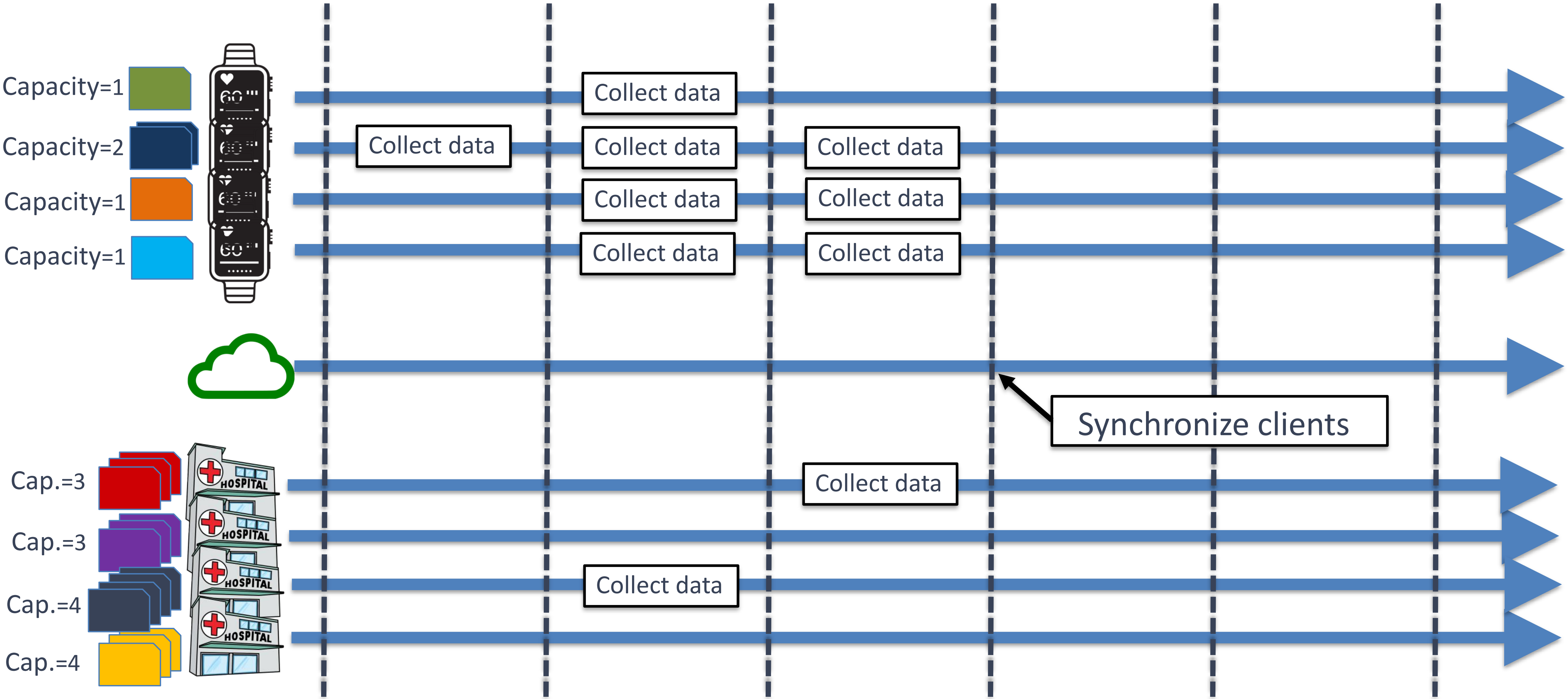
Federated Learning for Data Streams



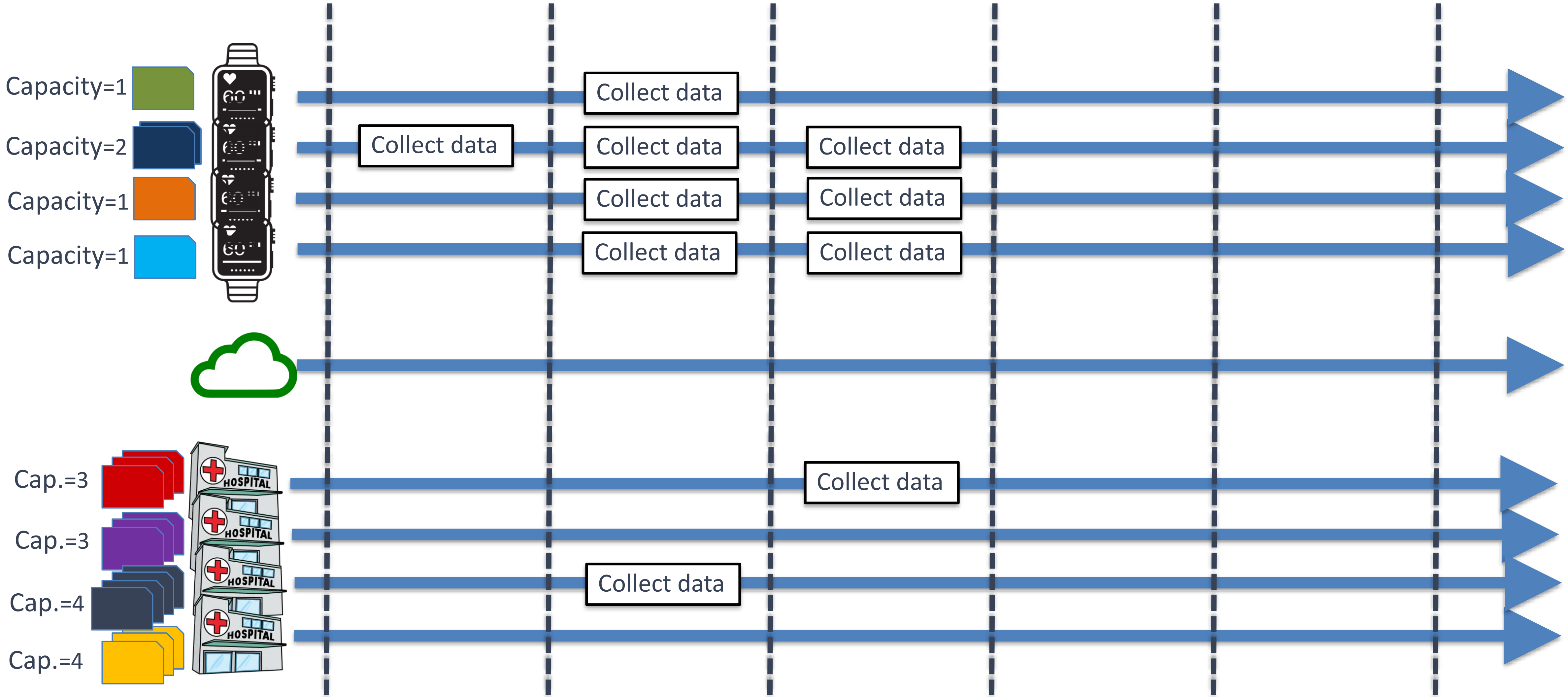
Federated Learning for Data Streams



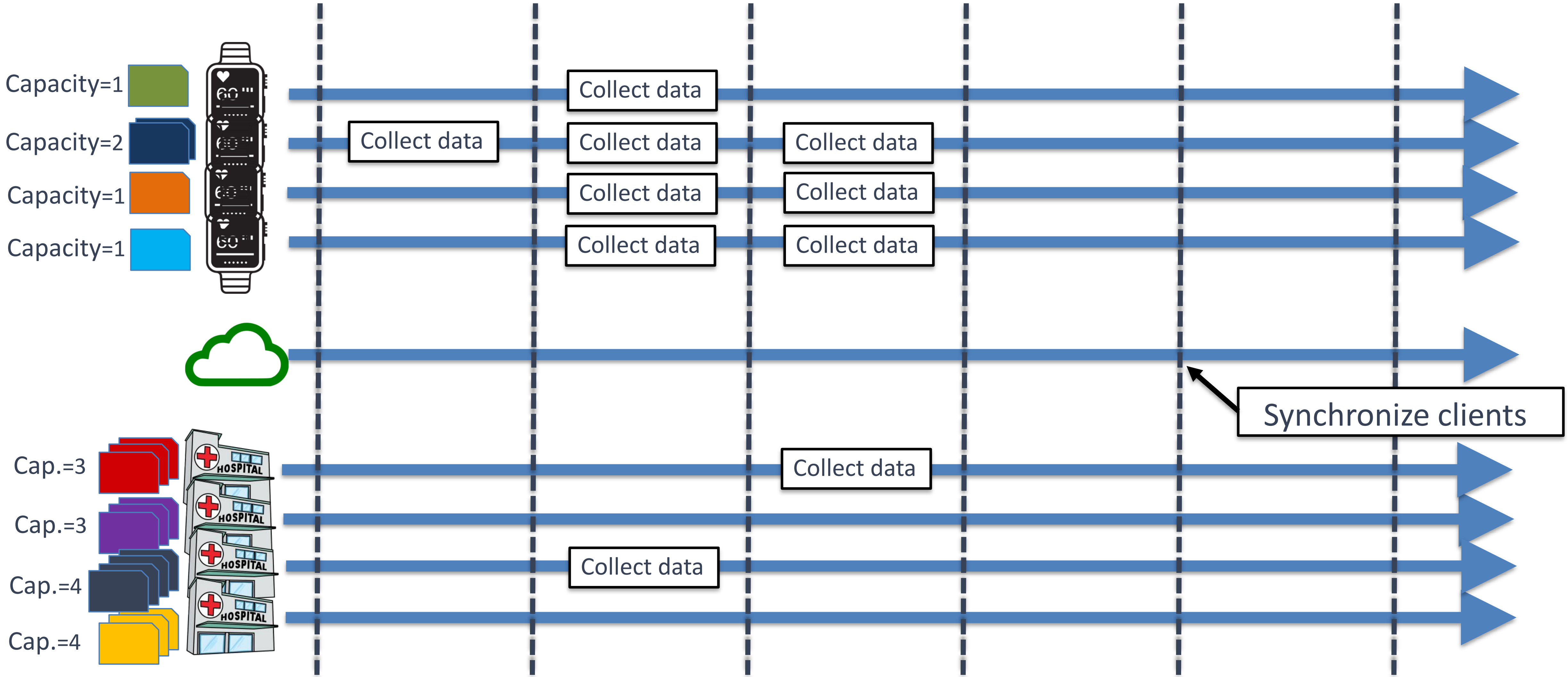
Federated Learning for Data Streams



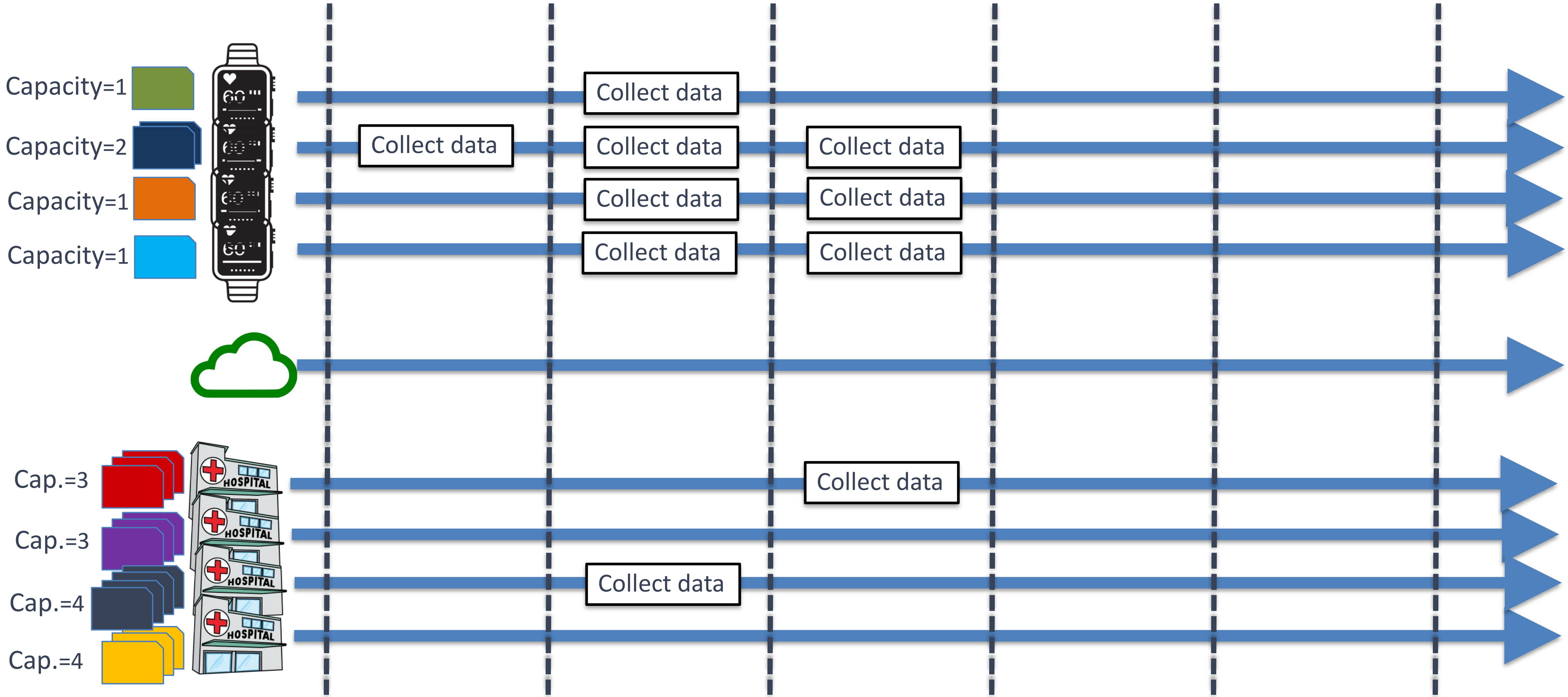
Federated Learning for Data Streams



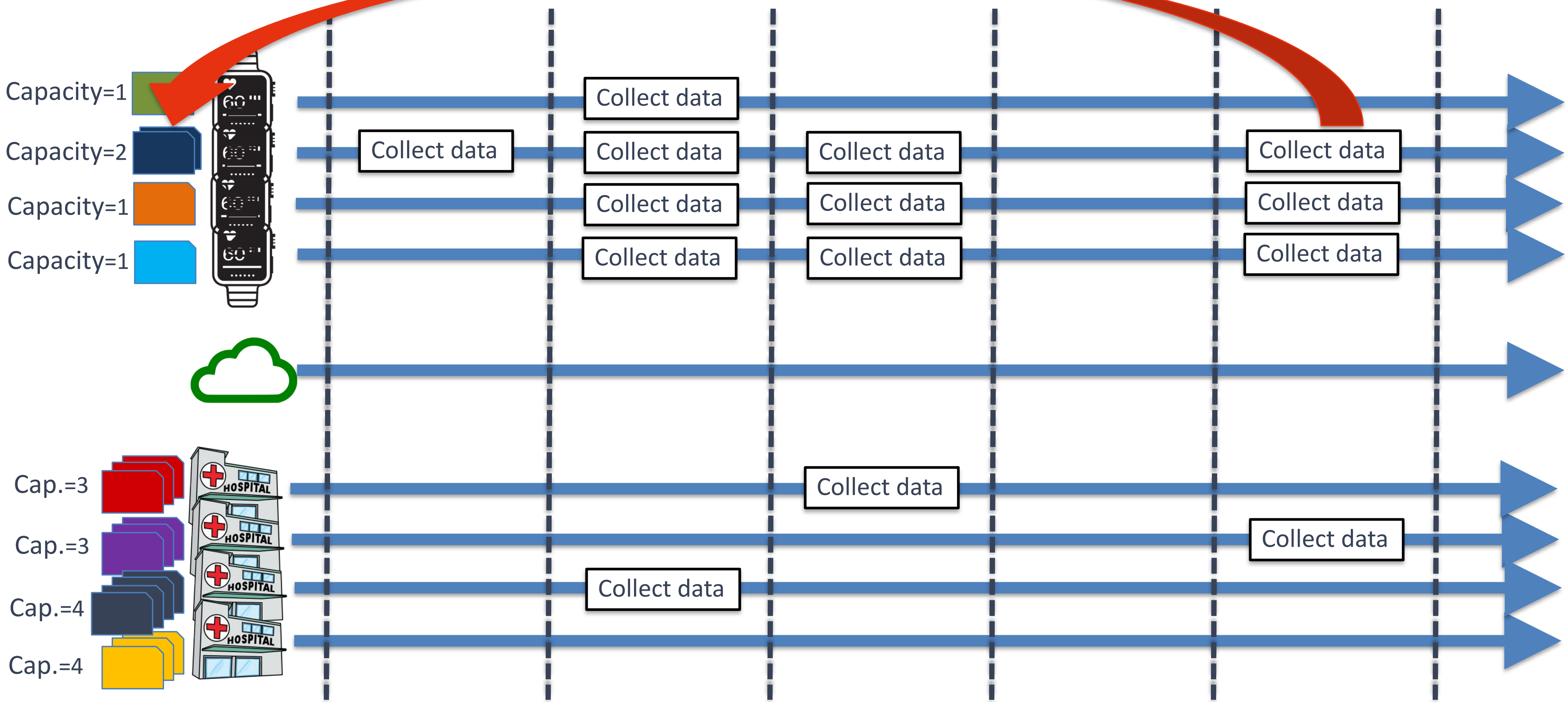
Federated Learning for Data Streams



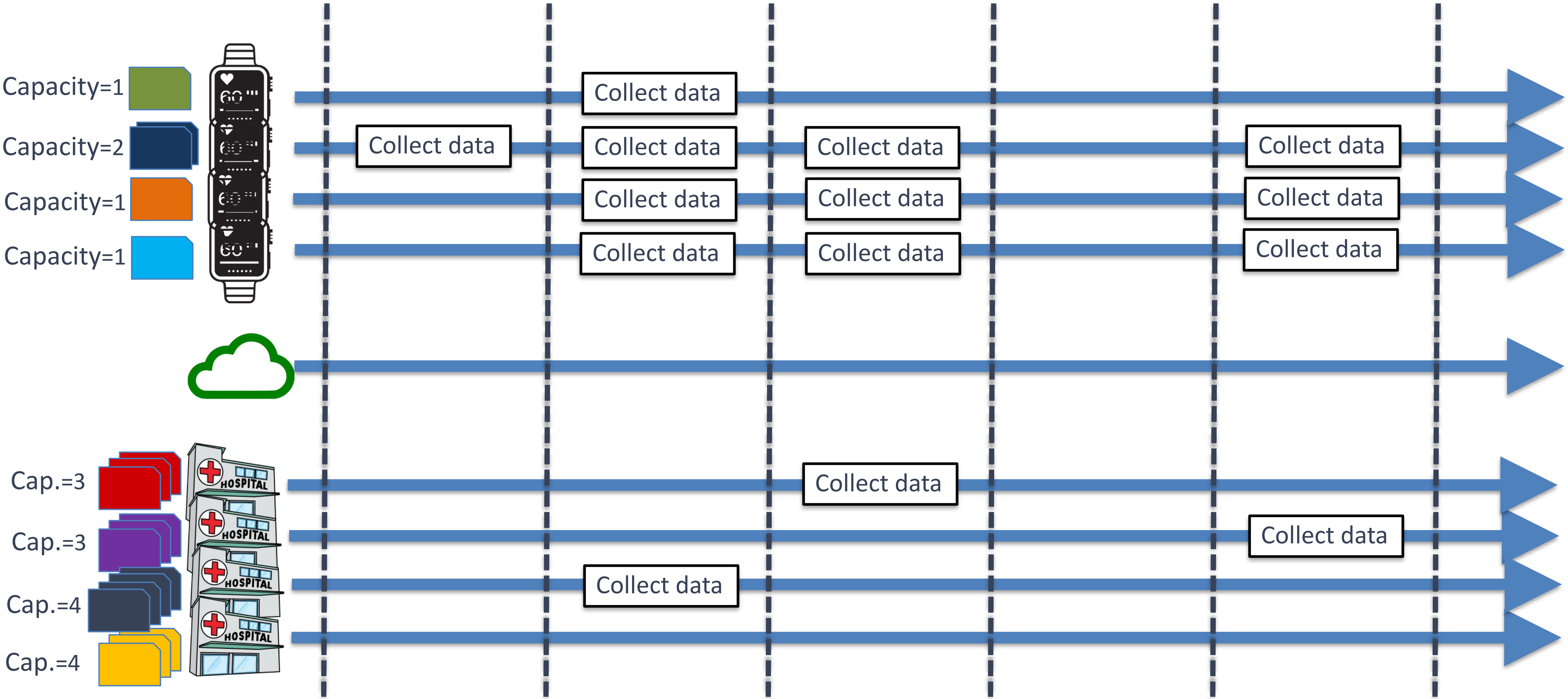
Federated Learning for Data Streams



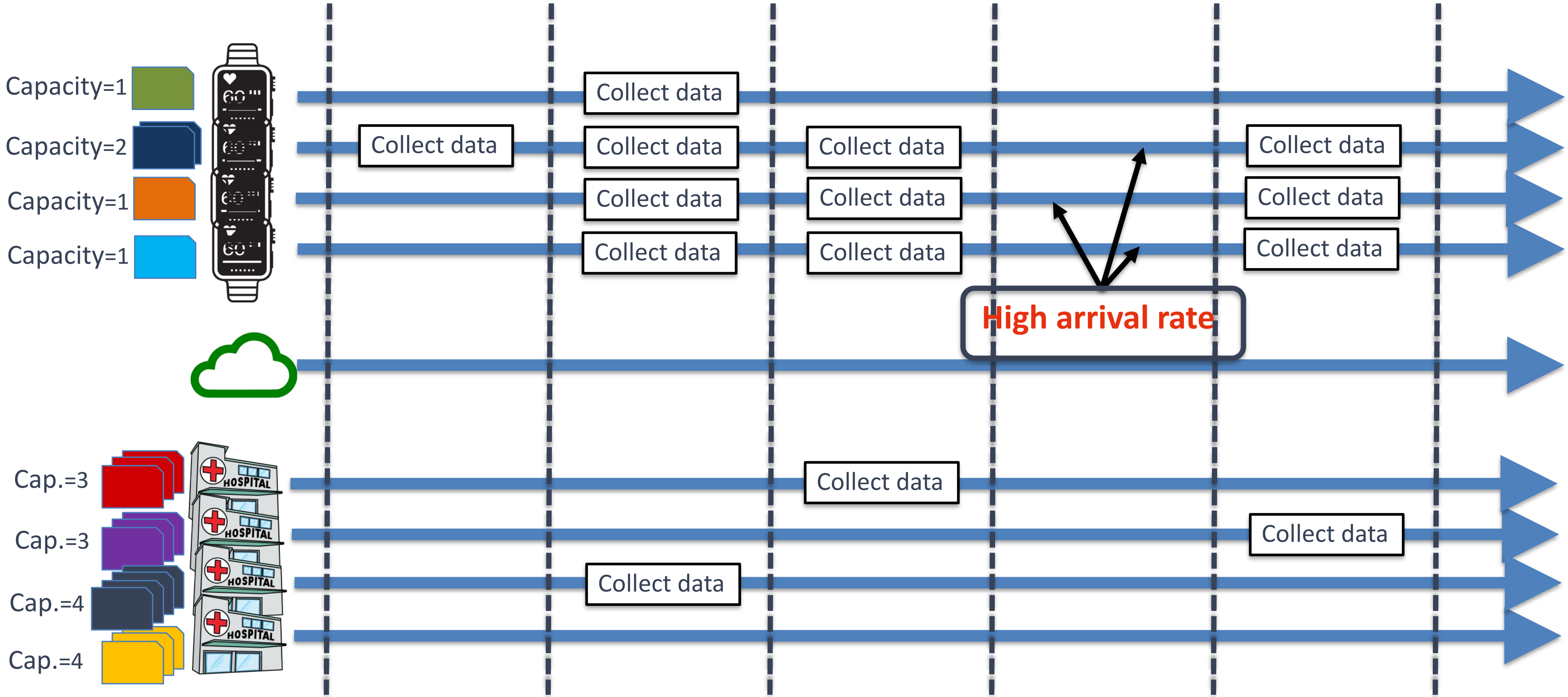
Federated Learning for Data Streams



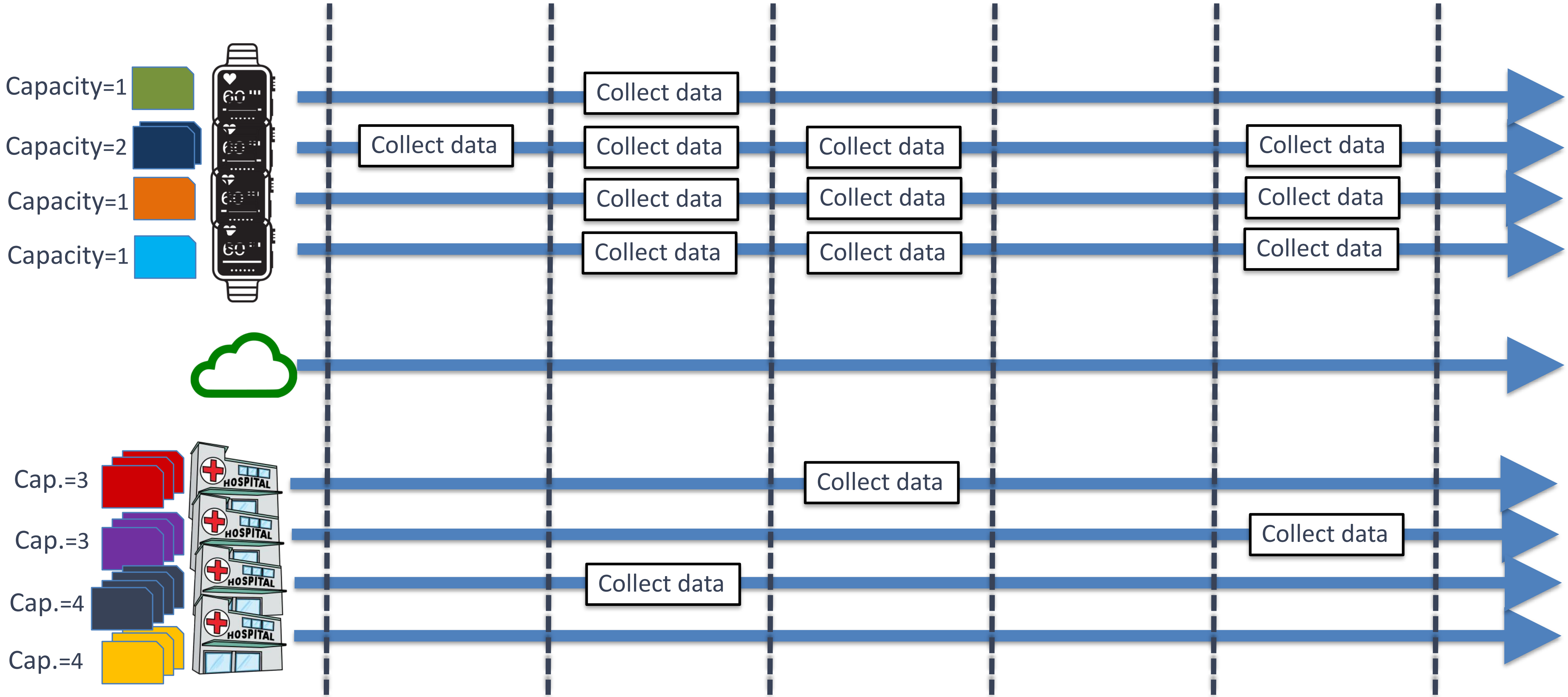
Federated Learning for Data Streams



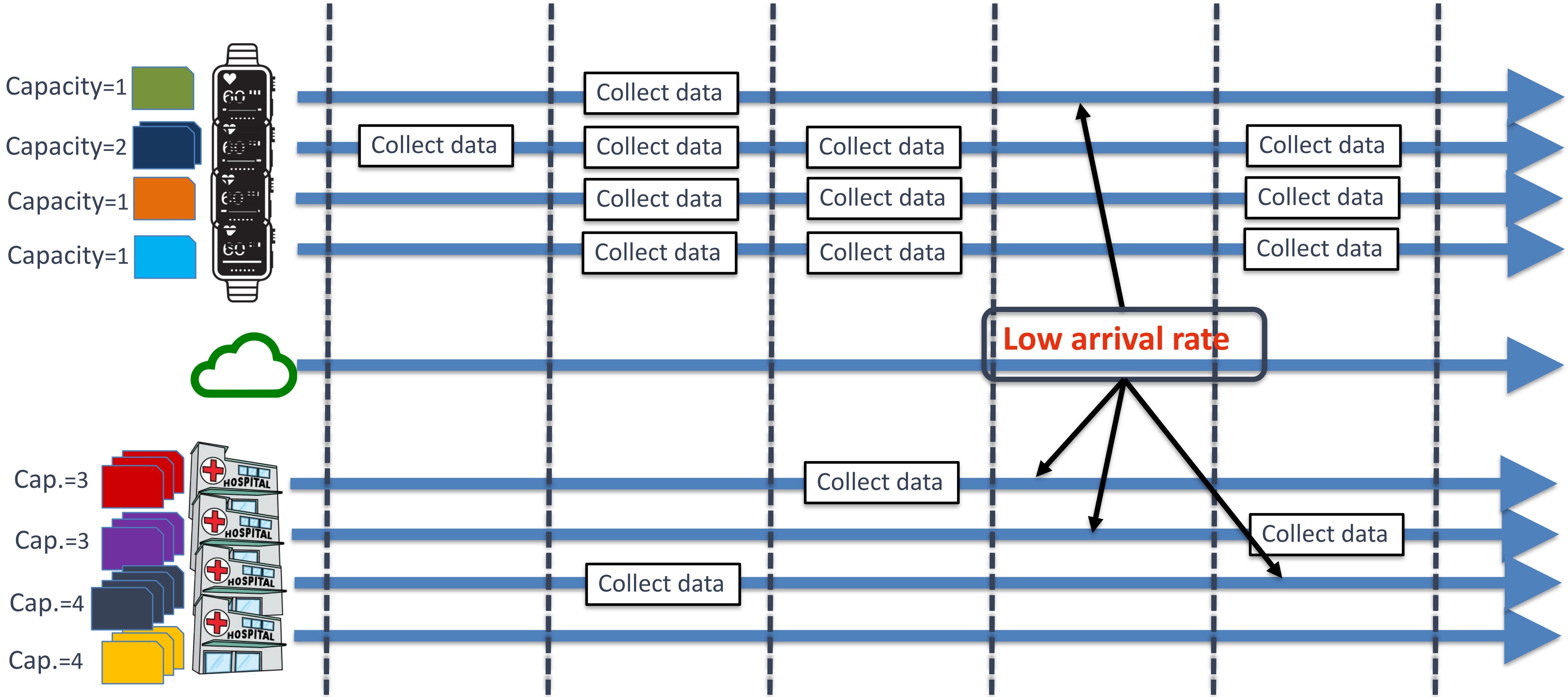
Federated Learning for Data Streams



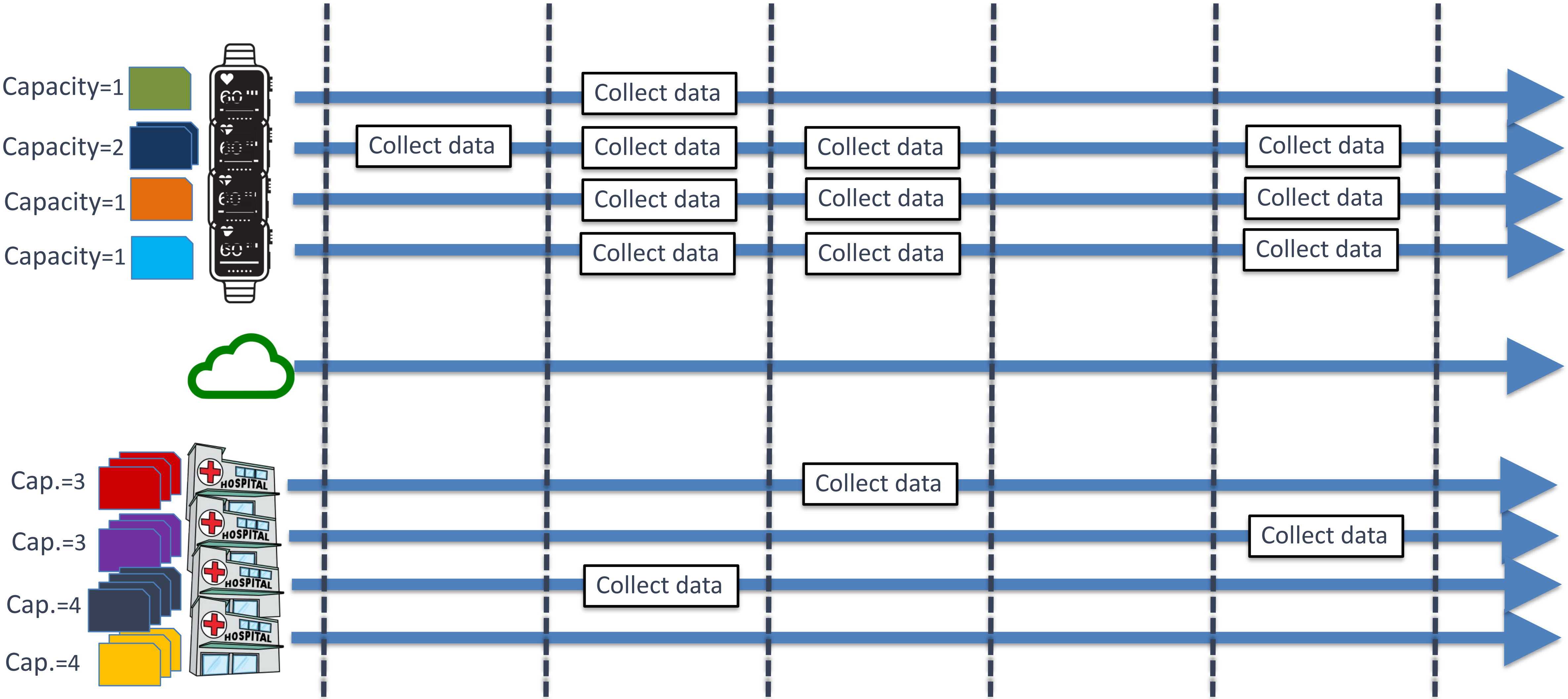
Federated Learning for Data Streams



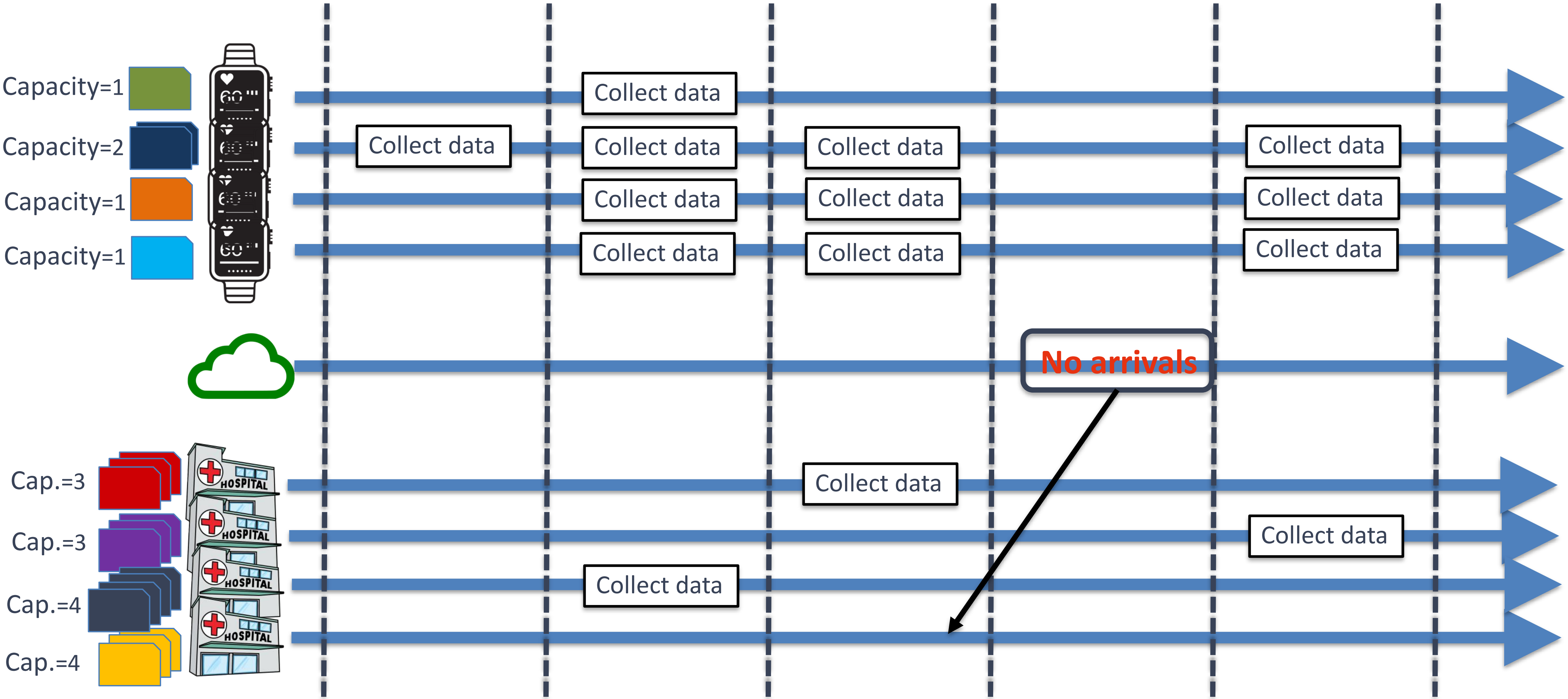
Federated Learning for Data Streams



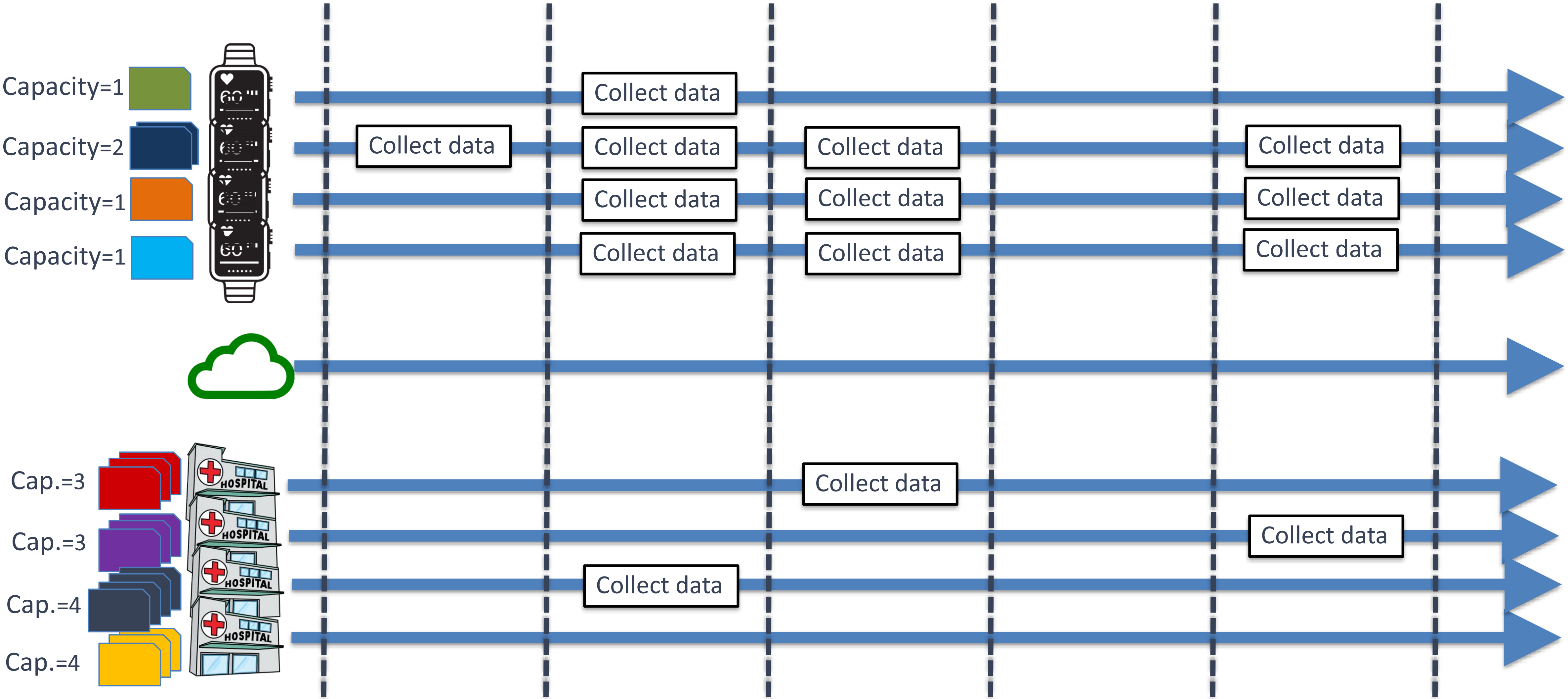
Federated Learning for Data Streams



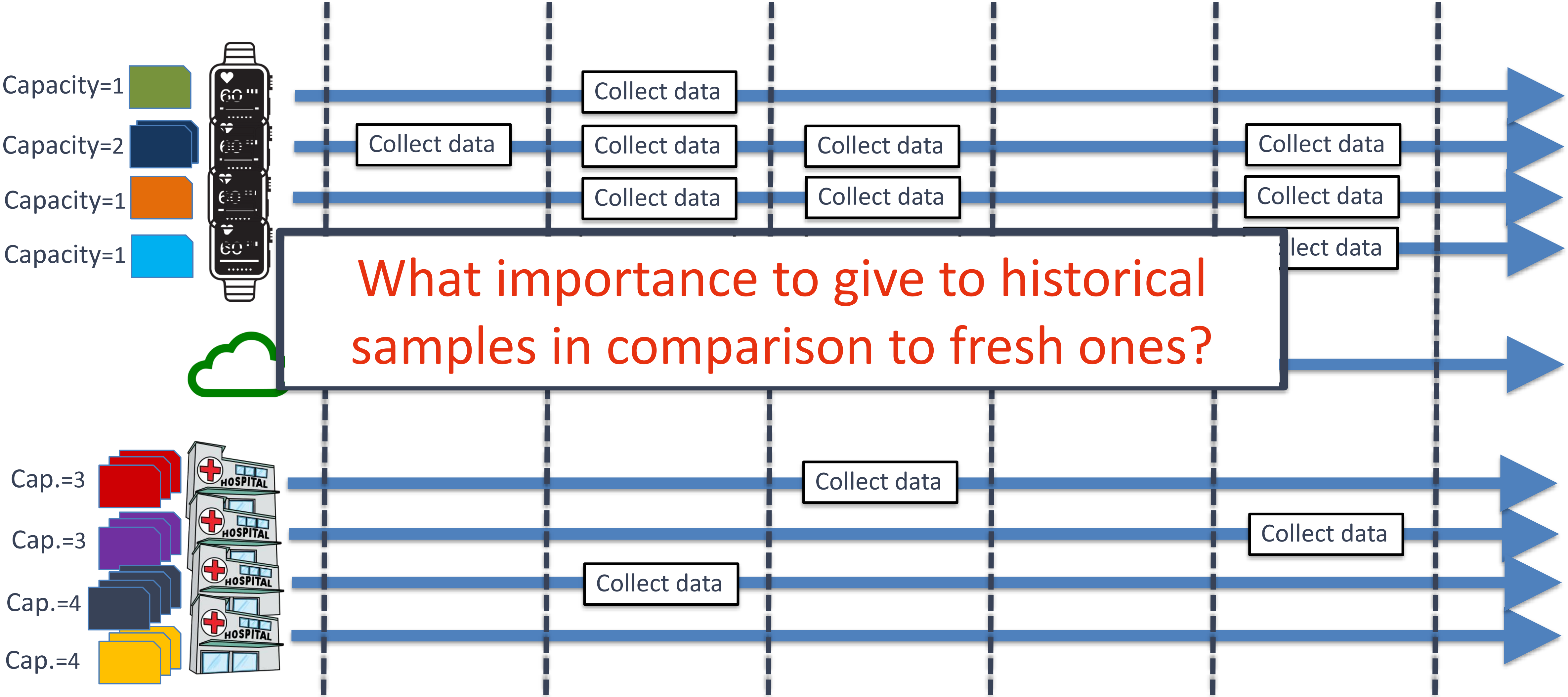
Federated Learning for Data Streams



Federated Learning for Data Streams



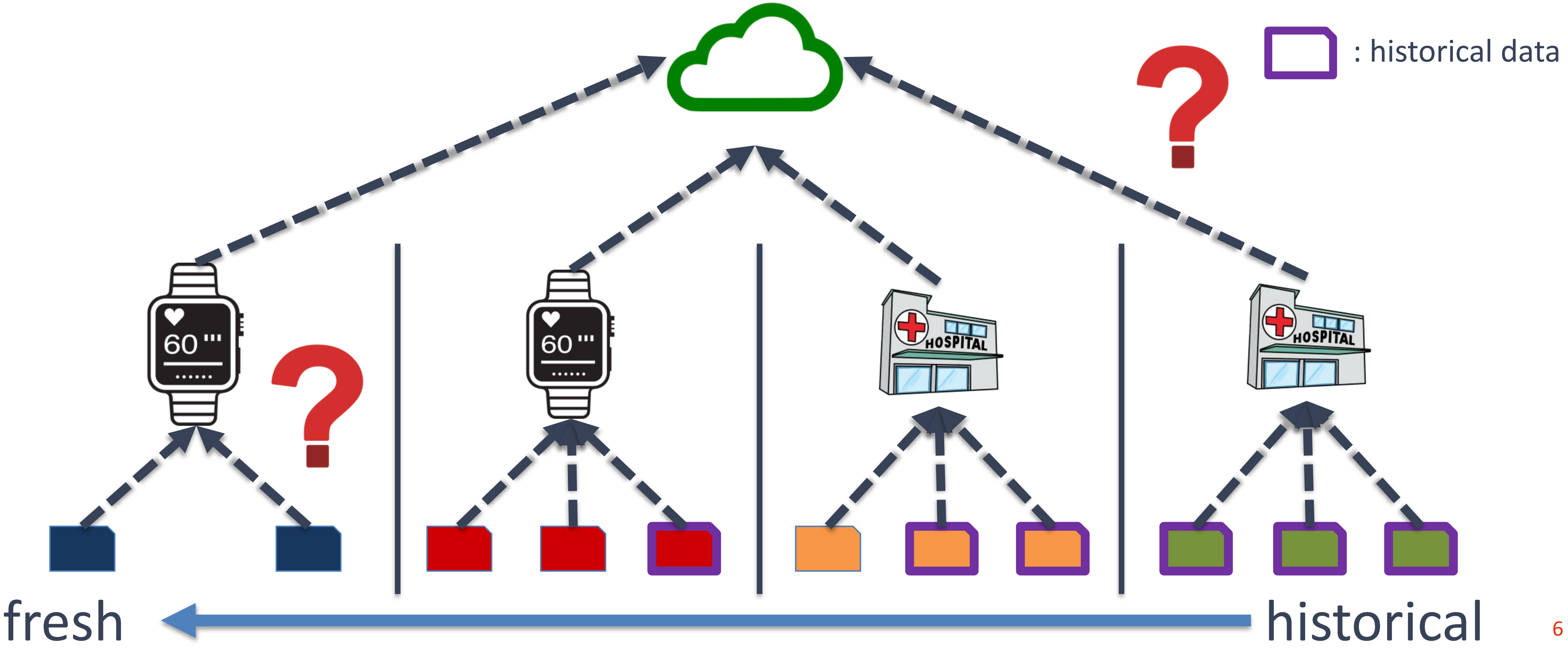
Federated Learning for Data Streams



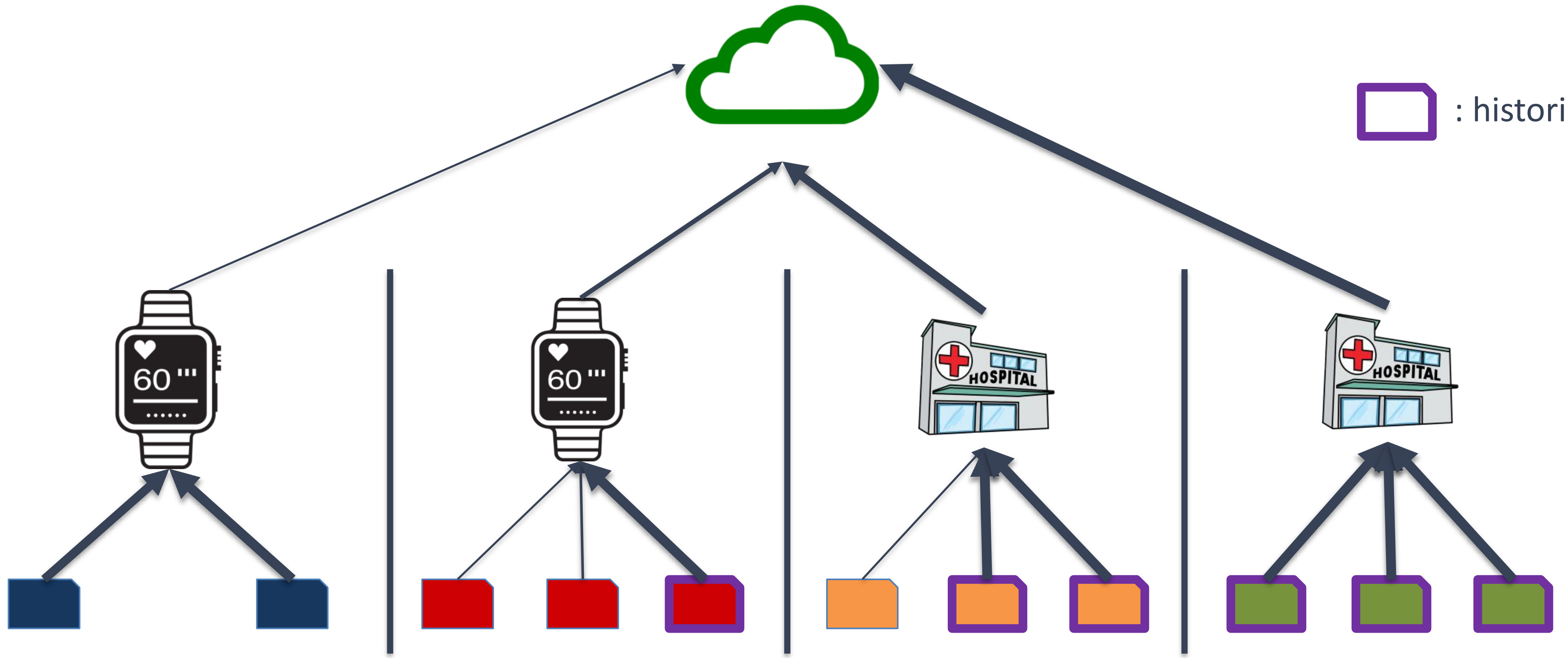
What importance to give to historical samples in comparison to fresh ones?

Federated Learning for Data Streams

What importance to give to historical samples in comparison to fresh ones?

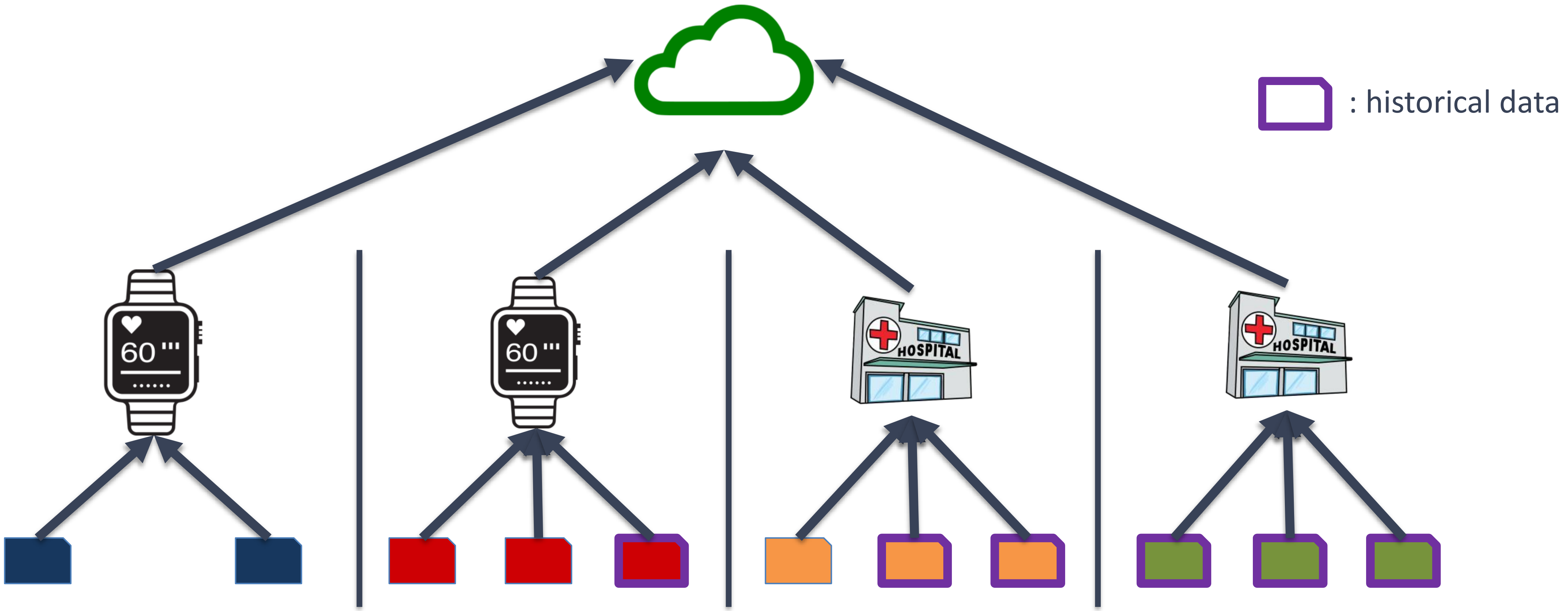


“Historical” strategy



: historical data

“Uniform” strategy



Theoretical Analysis

Reminder: error decomposition in ML

"True Error" = "Optimization Error" + "Generalization Error" + "Approximation Error"

"**Optimization Error**" is large when gradients are noisy

"**Generalization Error**" is large when few samples are available

"**Approximation Error**" only depends on the hypothesis space (model architecture)

Theoretical Analysis

"Historic"

- small variance
- small efficient number of samples
- low optimization error
- high generalization error

"Uniform"

- large variance
- large efficient number of samples
- high optimization error
- low generalization error

Theoretical Analysis

"Historic"

- sma
- sma

  samples
- low optimization error
- high generalization error

"Uniform"

- mber of

  samples
- high optimization error
- low generalization error

Is there a better compromise than these two extreme strategies?

Theoretical Analysis

Is there a better compromise than these two extreme strategies?

- ➢ The answer is "**usually YES**"

Theorem (informal)

\# clients

client importance

$$\text{True Error} \leq c_0 + c_1 \cdot \sqrt{\sum_{m=M_{\text{hist}}+1}^{M} p_m^2} + c_2 \cdot \sqrt{\sum_{m=1}^{M} \frac{p_m^2}{n_m}}.$$

\# historical clients

relative size

Practical Algorithm

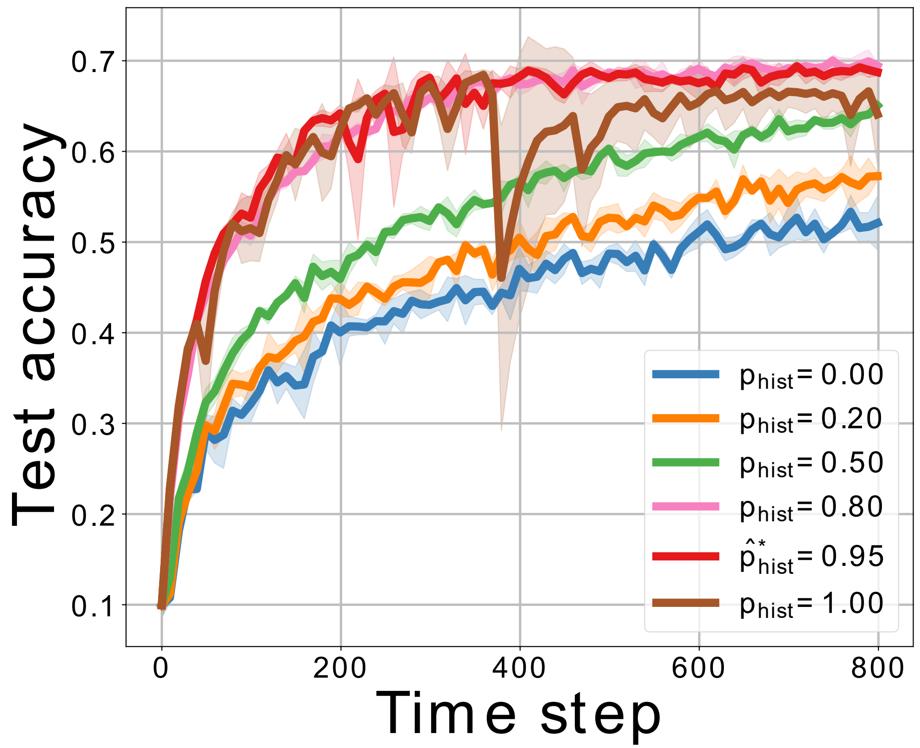
1. estimate the constants on historical data

2. optimize the upper-bound

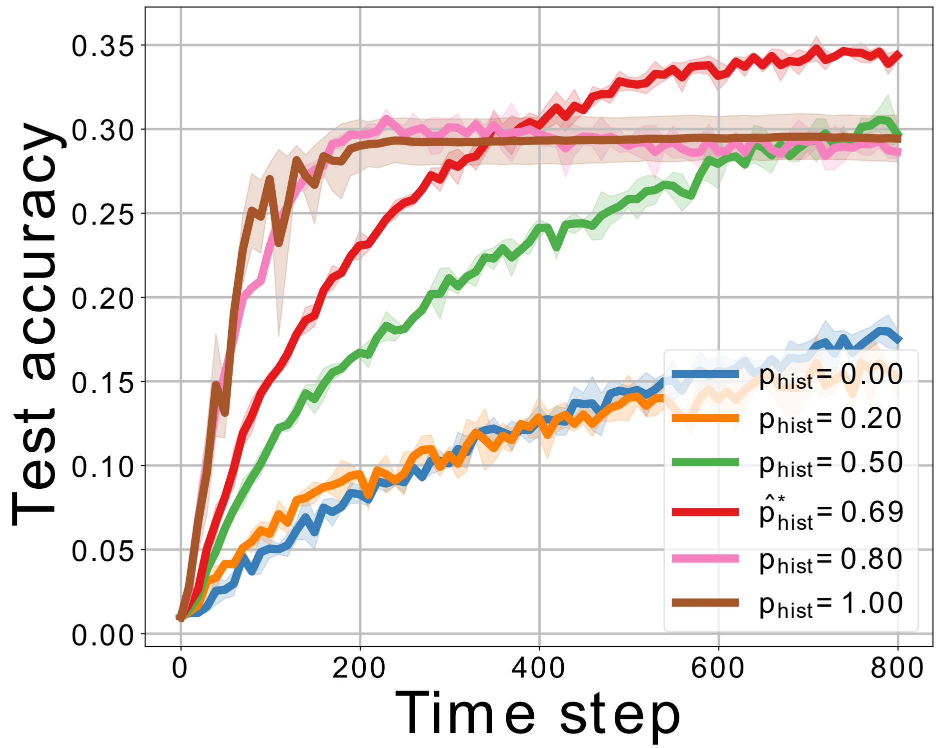
3. run federated averaging with the resulting weights

Experimental Results

Scenario with large number of historical samples (50%)



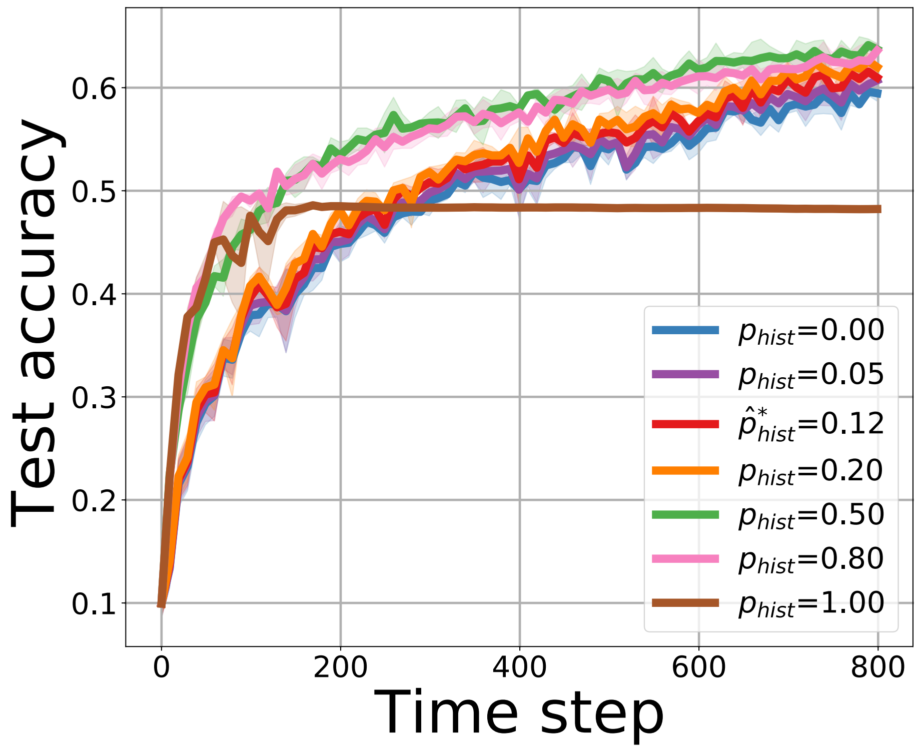
CIFAR-10



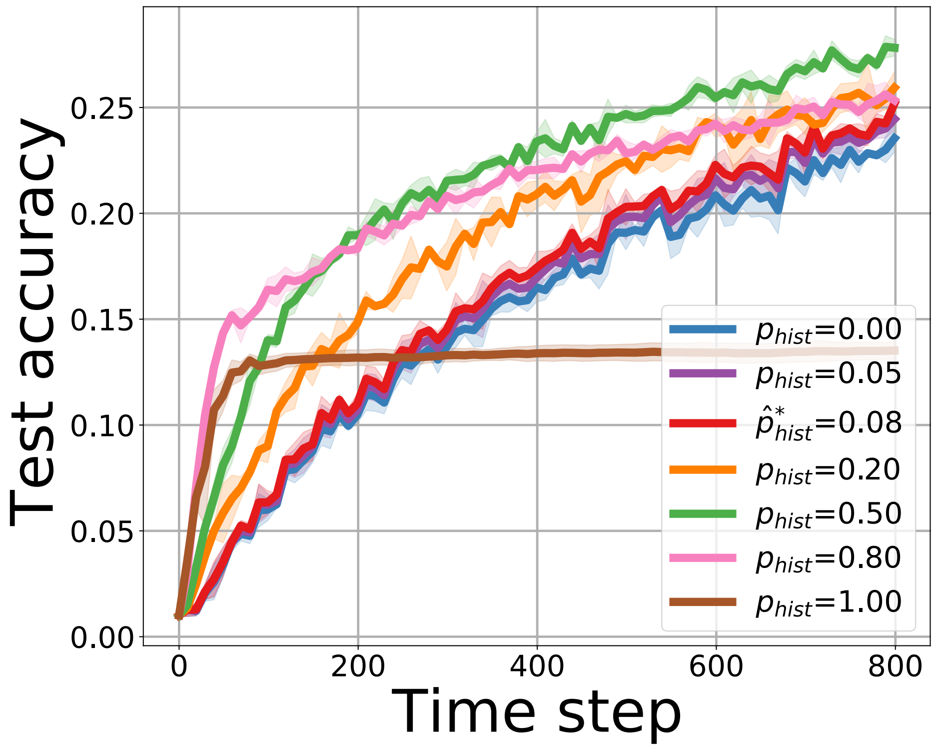
CIFAR-100

Experimental Results

Scenario with a few historical samples (5%)



CIFAR-10



CIFAR-100

Conclusion

1. Formalize the problem of federated learning for data streams
2. Theoretical analysis reveals a new generalization-optimization tradeoff
3. Practical algorithm to decide the relative importance samples

Questions