

# Secure P2P Federated Learning

Leonardo Randacio

# Federated Learning

Una buona idea con un grosso ostacolo

# Secure Average Computation

Una possibile soluzione

con n numero di peer e m numero di pesi

- partiziona pesi
- scambio pesi partizionati
- salva pesi ricevuti  $pwt$  ( $|pwt| = n \times m$ )
- calcola somme parziali  $pswt$  ( $|pswt| = m$ )

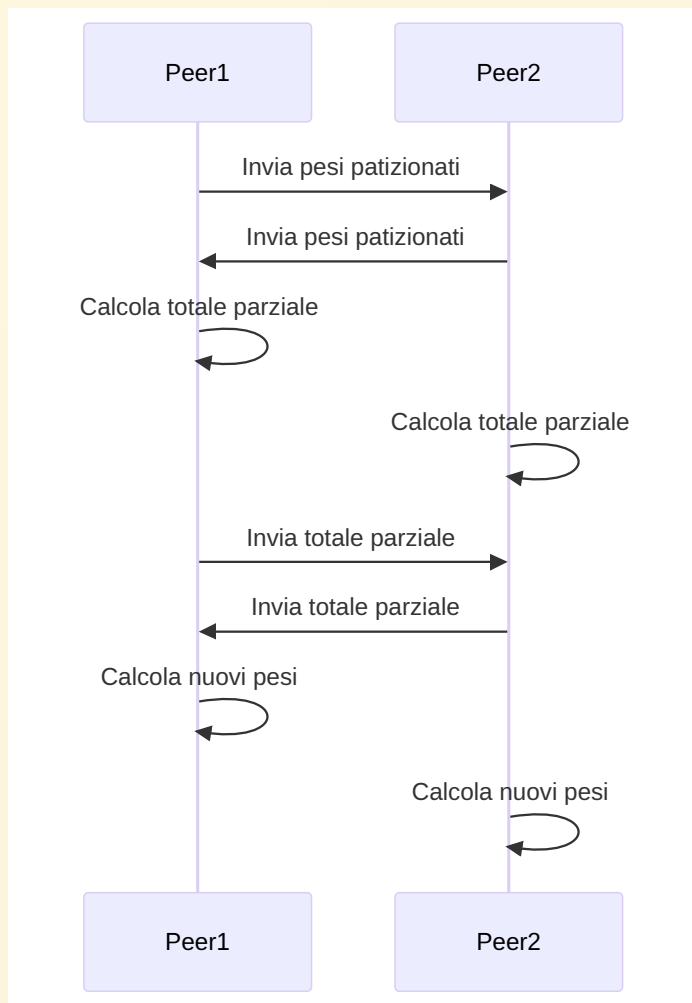
$$pswt = \sum_{i=1}^n pwt_i$$

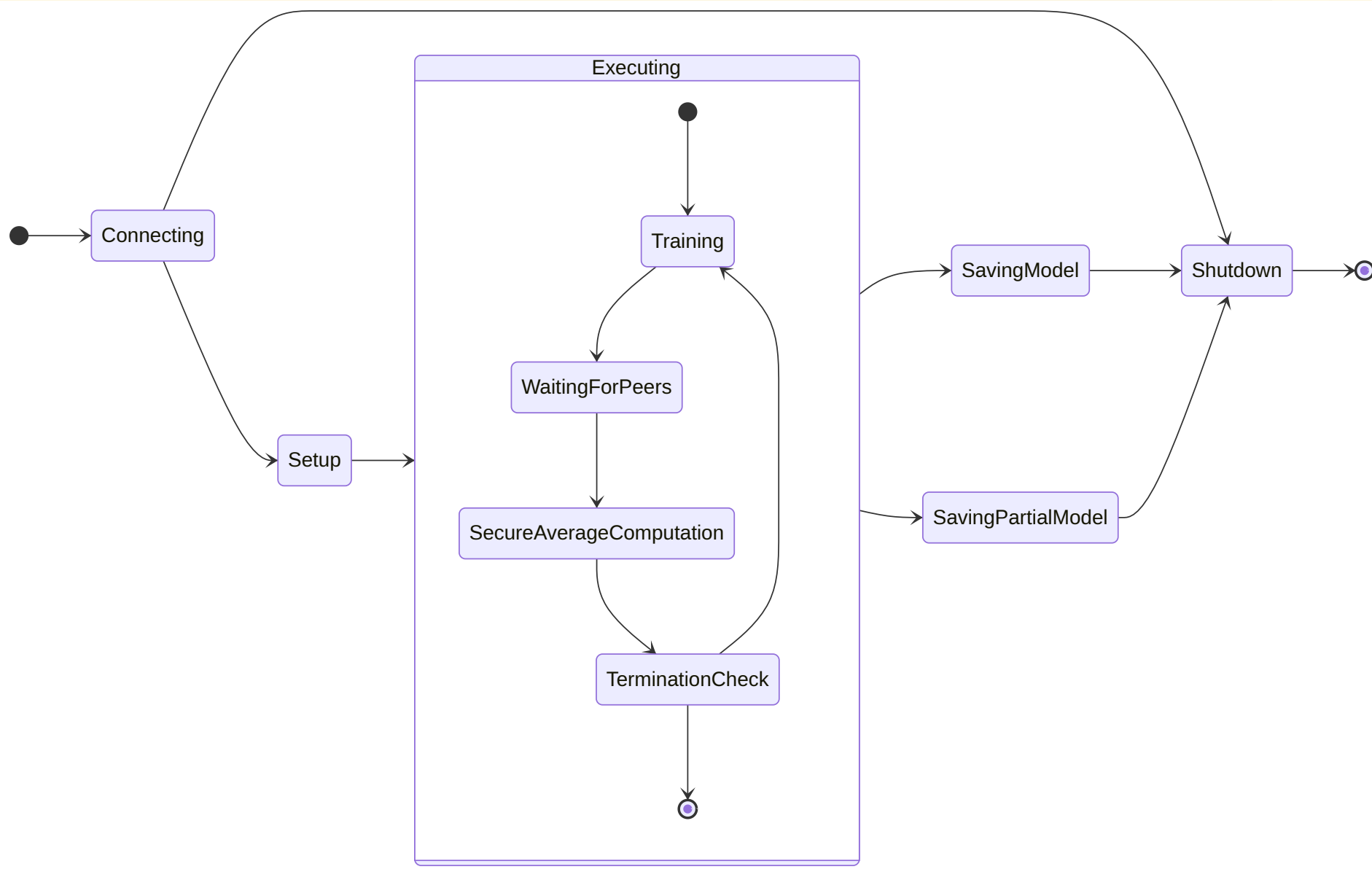
Continua...

con  $n$  numero di peer e  $m$  numero di pesi

- calcola somme parziali  $pswt$  ( $|pswt| = m$ )
- scambio somme parziali
- salva somme parziali ricevute  $swt$  ( $|swt| = n \times m$ )
- calcola nuovi pesi  $wt$  ( $|wt| = m$ )

$$wt = \frac{1}{n} \sum_{j=1}^n swt_j$$





# Messaggi

Con  $n$  numero di peer,  $r$  numero di epoche

- Pesi iniziali =  $n - 1$
- Ogni epoca: =  $3n(n - 1)$

Messaggi inviati =  $(1 + 3rn)(n - 1) = O(n^2r)$

(Senza contare heartbeat)

# Risultati

| Numero di<br>Peer | Numero di<br>round | Test<br>accuracy | Tempo di<br>esecuzione |
|-------------------|--------------------|------------------|------------------------|
| 2                 | 9                  | 0.88             | 4m 53s                 |
| 3                 | 7                  | 0.88             | 5m 5s                  |
| 4                 | 16                 | 0.88             | 7m 50s                 |
| 5                 | 16                 | 0.88             | 8m 45s                 |
| 6                 | 16                 | 0.88             | 9m 11s                 |
| 7                 | 23                 | 0.88             | 12m 23s                |

# Future Work

- Leader election
- Gossip heartbeat
- Dinamic members and recovery
- Length-prefixed framing
- Bad actor simulation
- Python library