

# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$

client



shard 1



shard 2

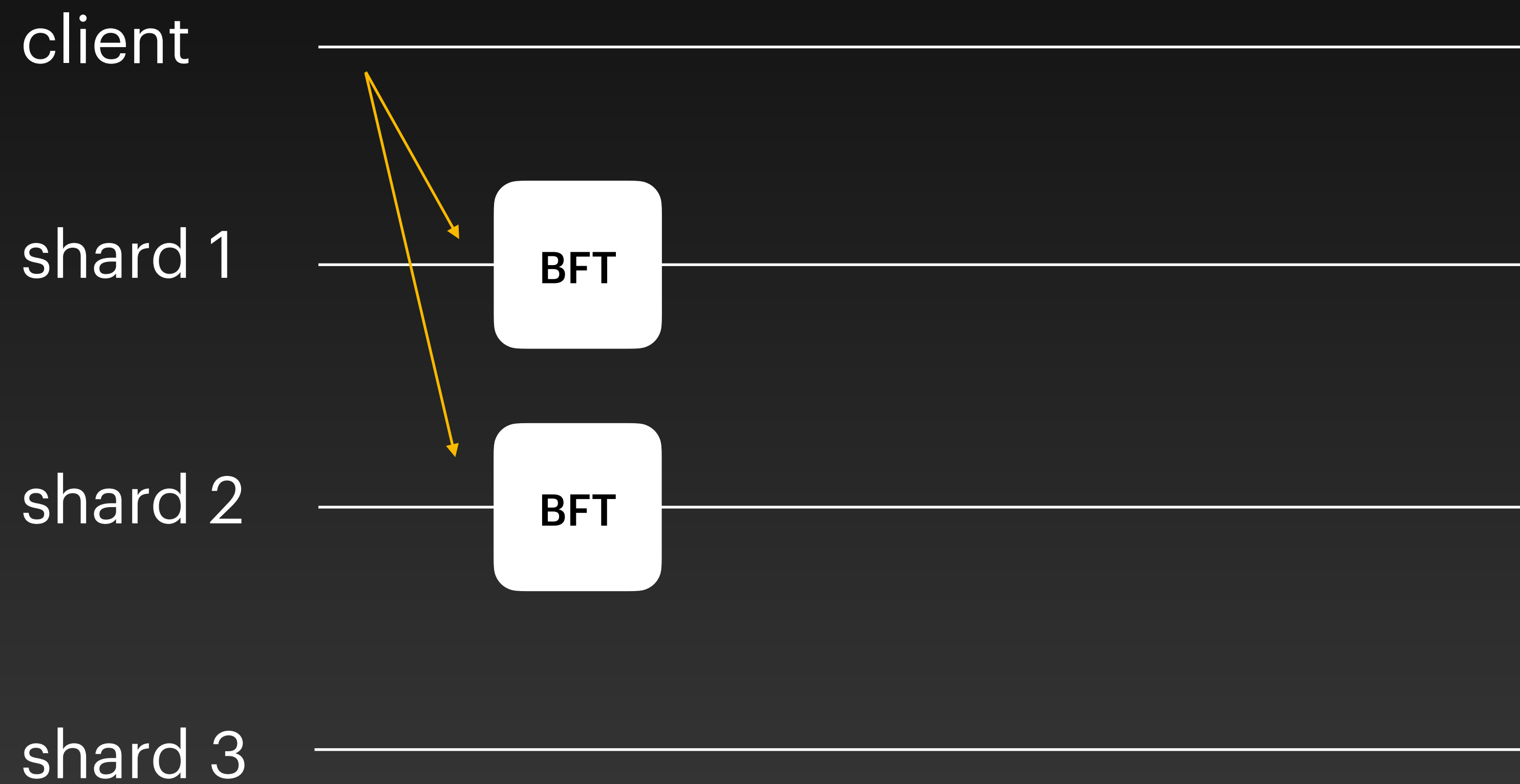


shard 3



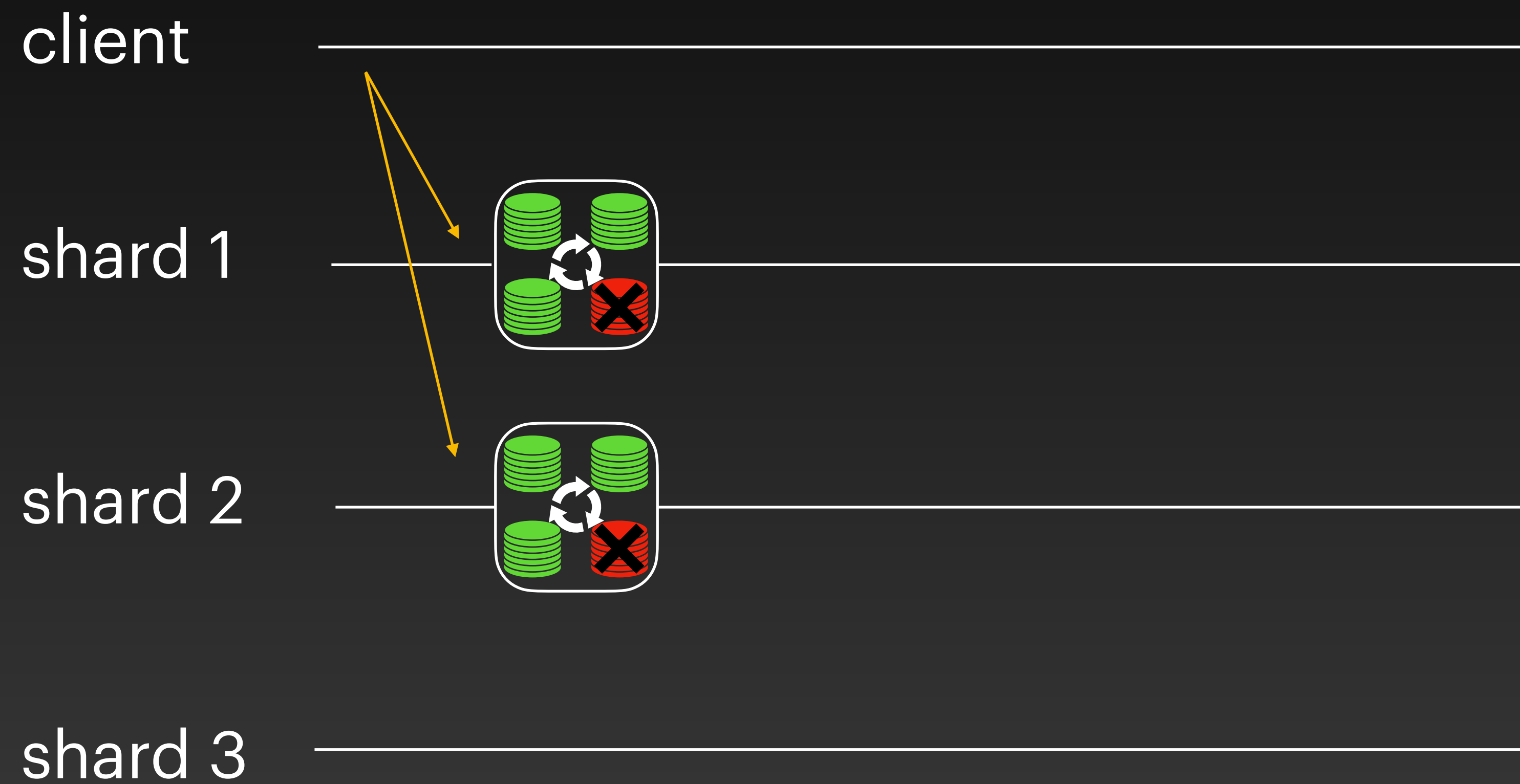
# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



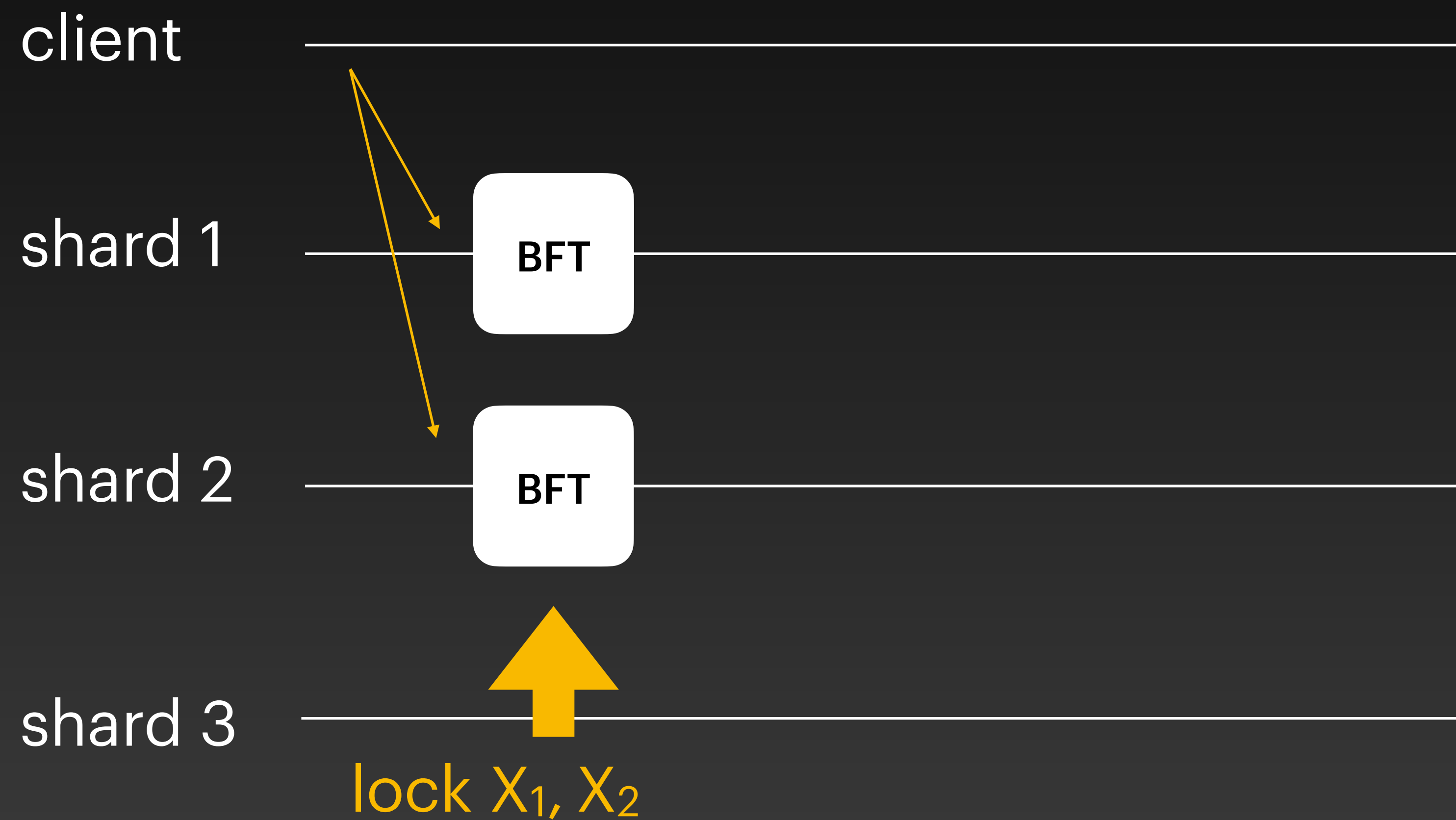
# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



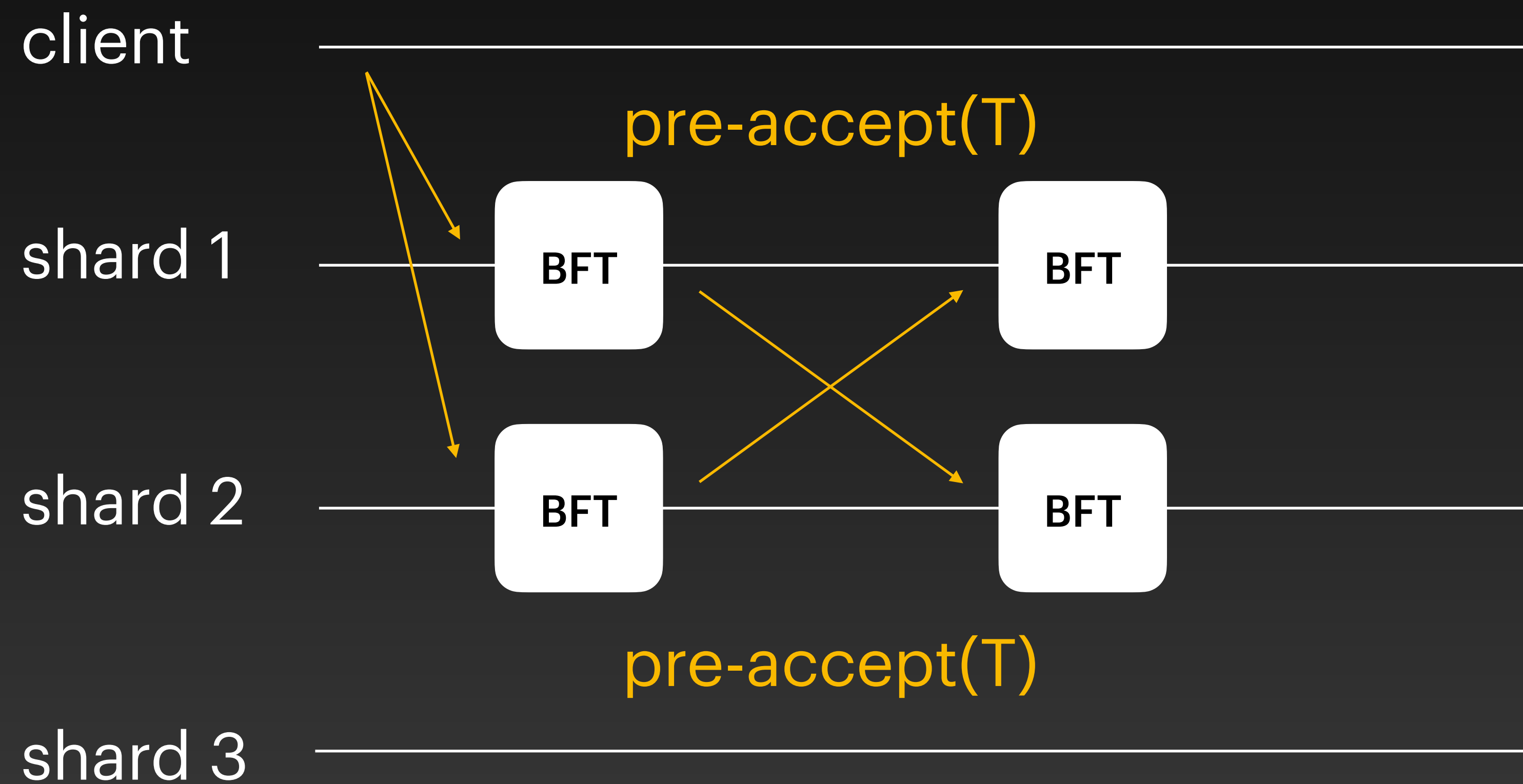
# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



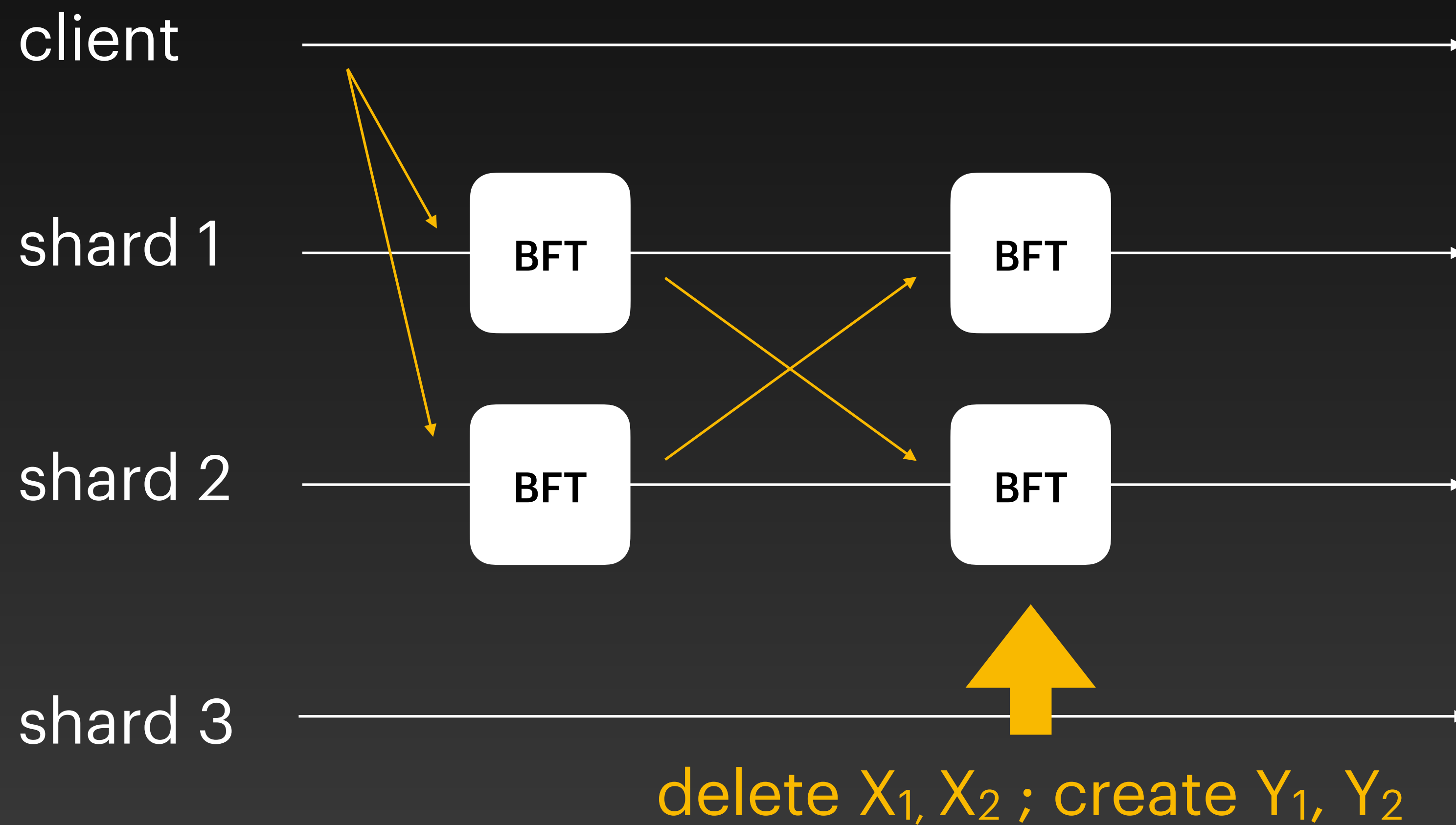
# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



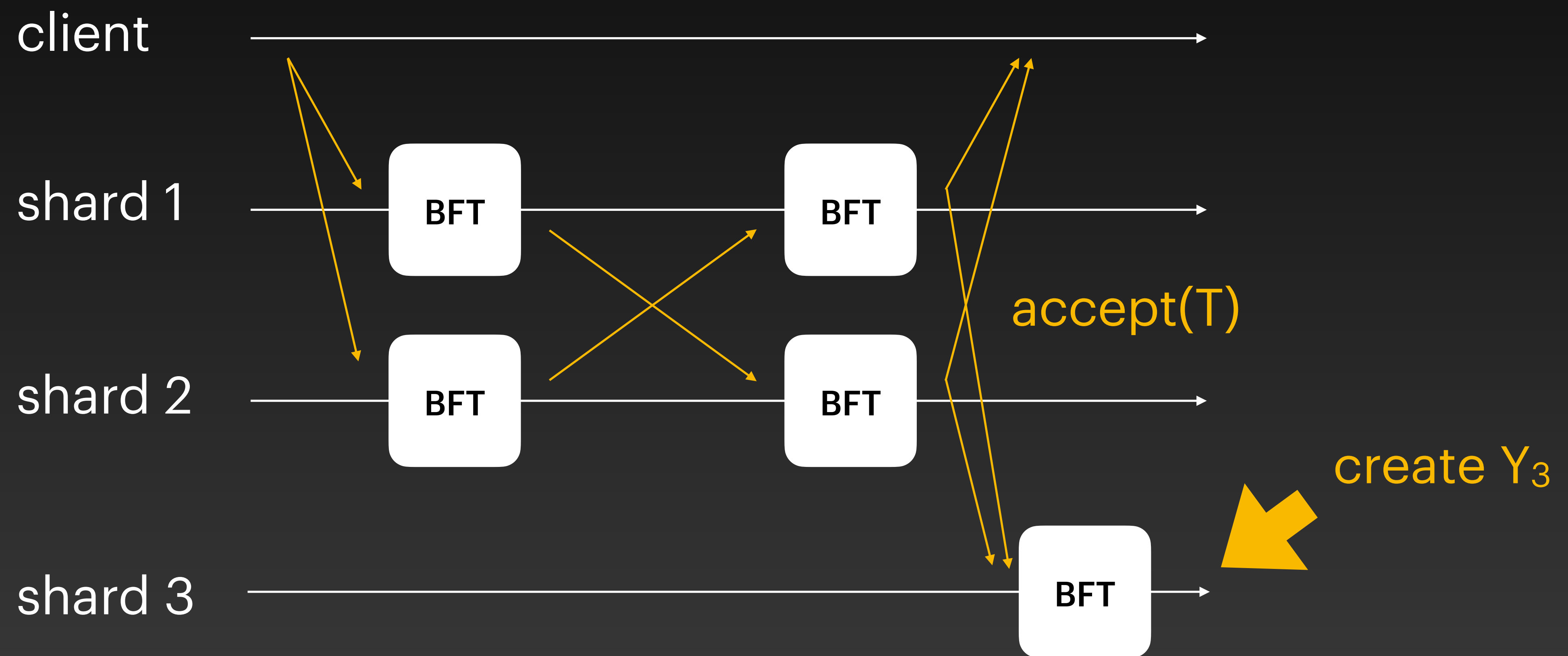
# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



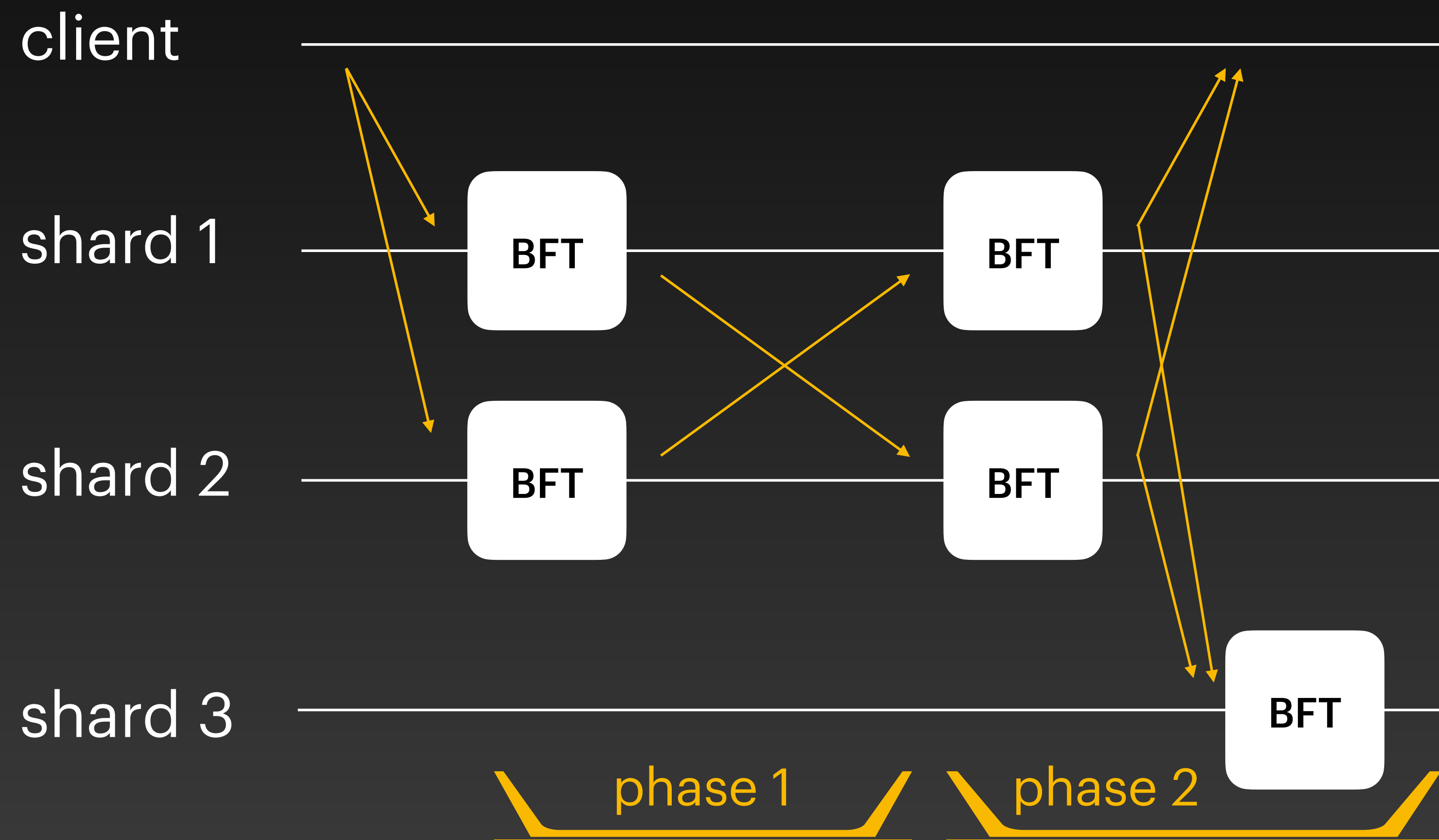
# S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



# S-BAC

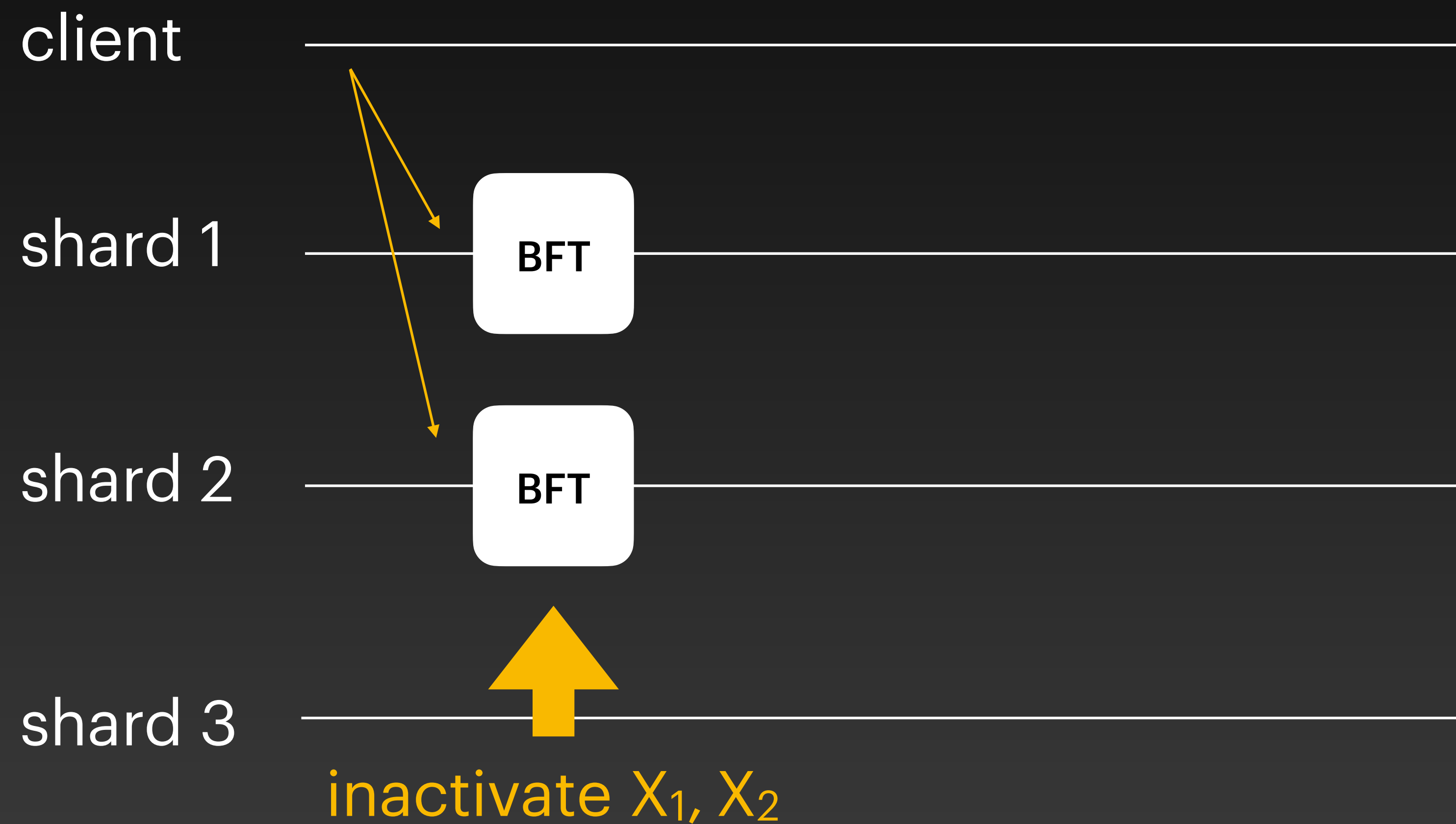
$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$





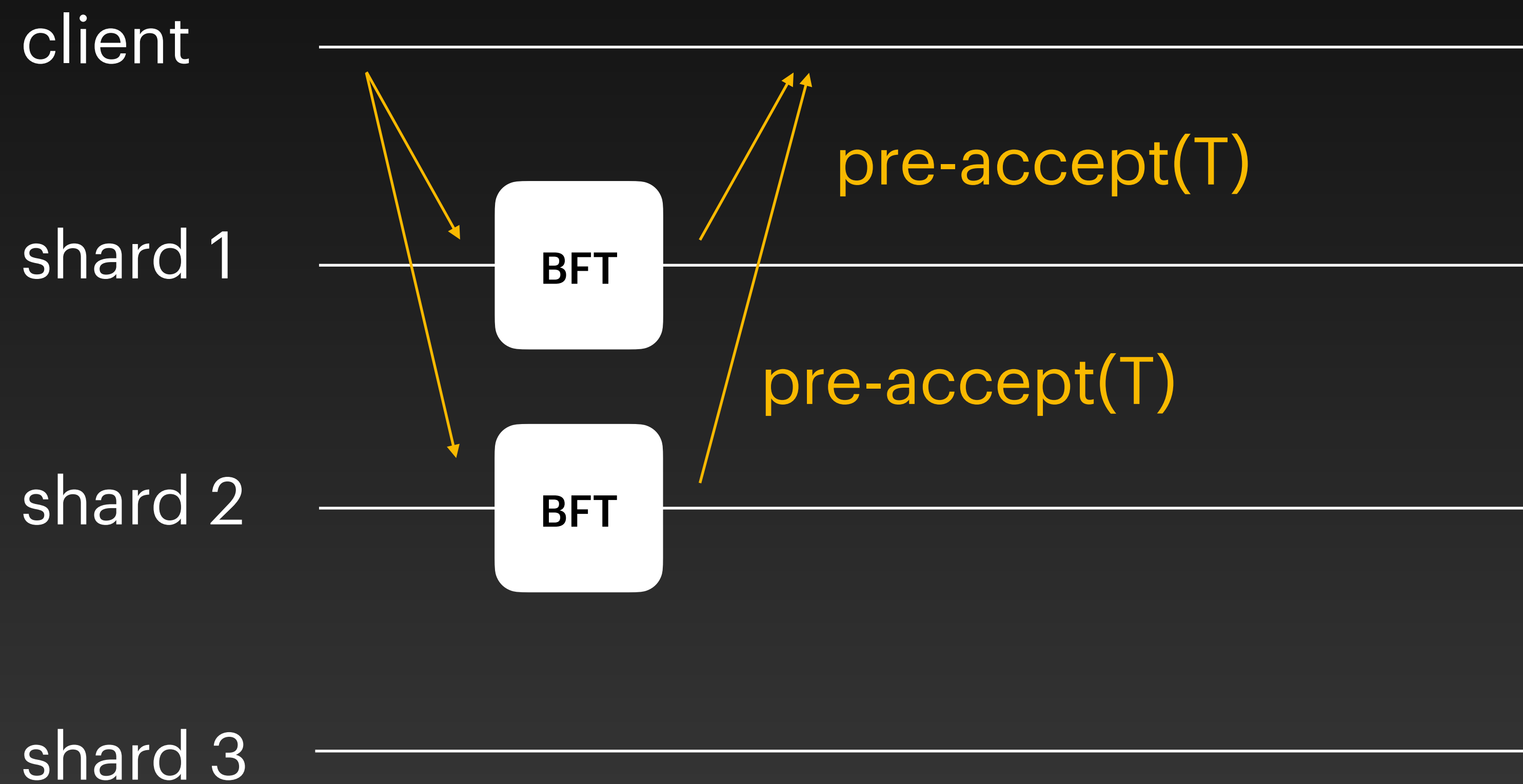
# Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



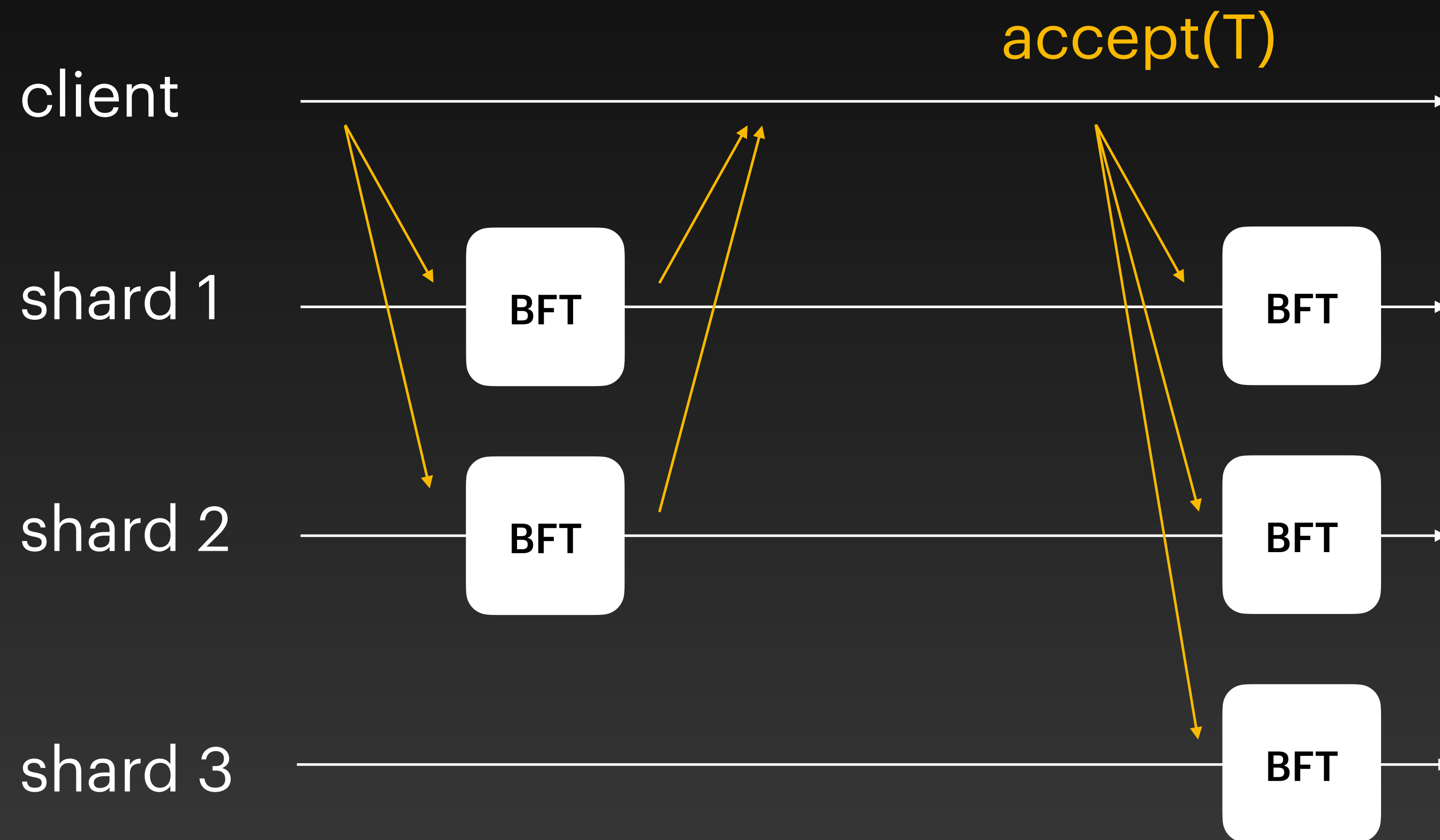
# Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



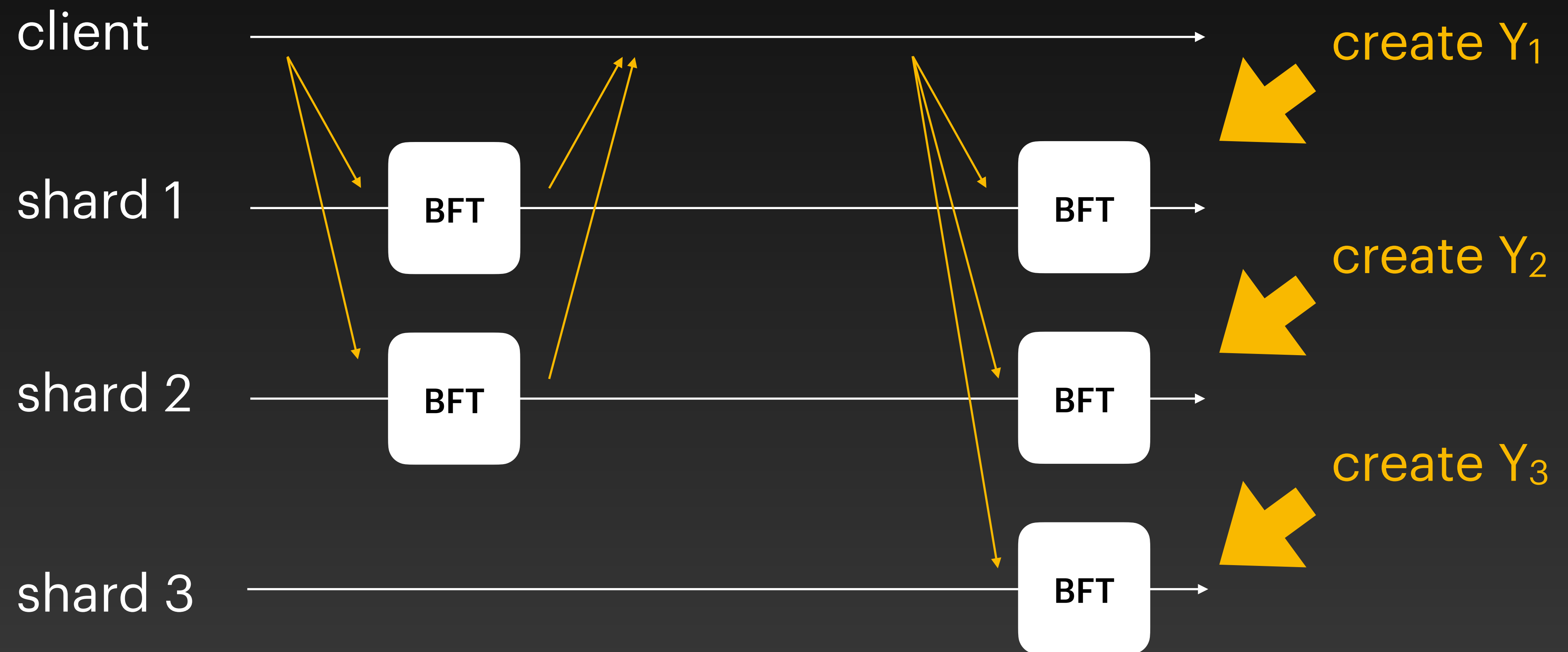
# Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



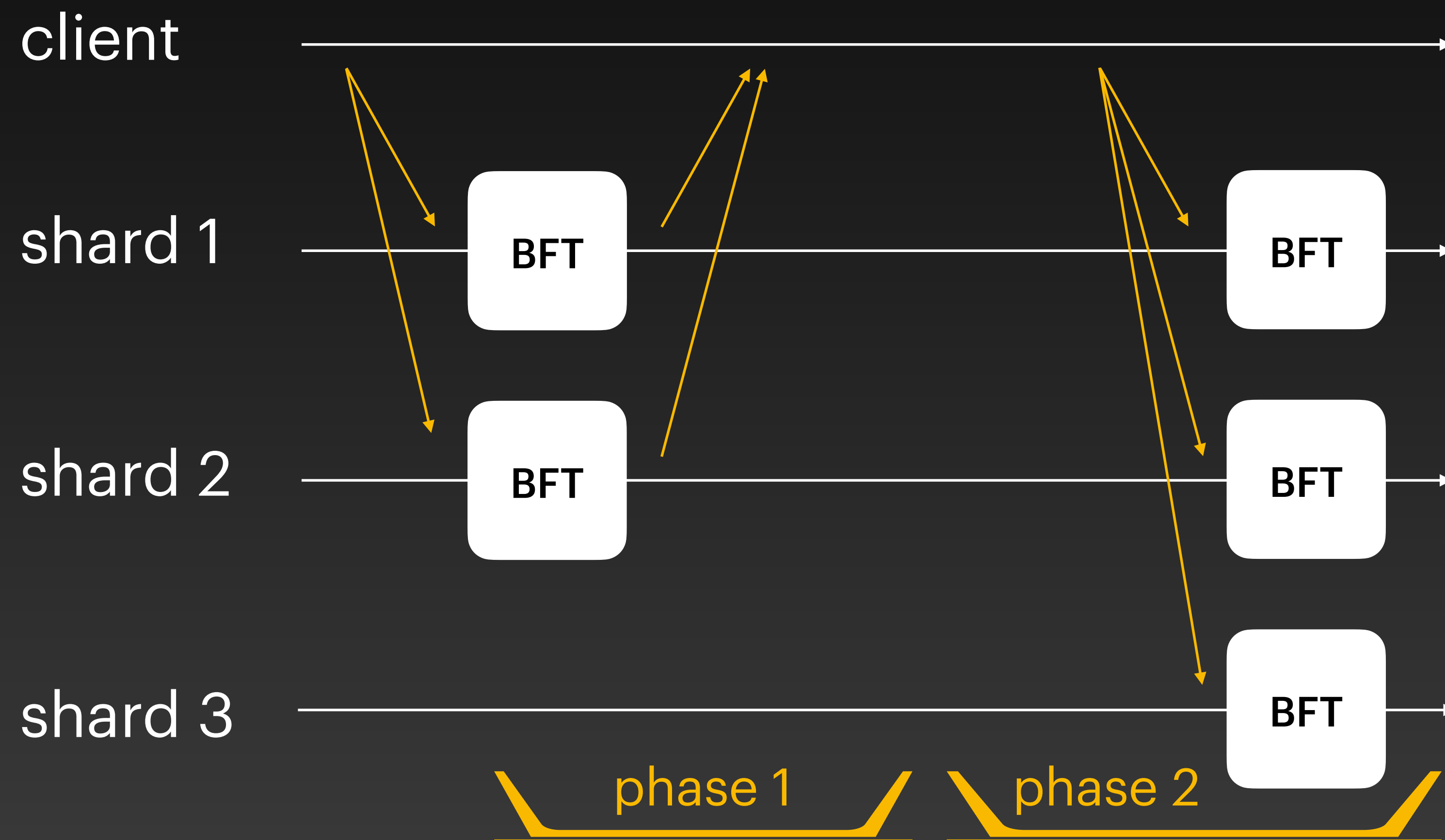
# Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



# Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



# Cross-Shard Consensus

How does it achieve linear scalability?

