NFS → **stateless, not scalable**, 

- failure issues - Good
- semantics """" - bad.

**CLIENT** → **SERVER**

must be complete

$\not$ not dependent on history
how to fix it?

Serious bodily
injury

too many strikes
cause

not searchable

Good

Does it matter

Server failure (cost)

Server failure (cost)
Increasing Scalability

- client caching reduces server overhead
- multiple servers for same thing
  
  processes or threads
  (multithreading)

- partition the files

- replication (??)
Semantics issues

Concurrent access

-> Concurrency not the happen (i)

Semantics issues

Coherence

-> Assume not to happen (i)
For a location file, should be model layer. NFS is not distributed.

What is a distributed ES?
1. Security/Protection
2. Scalability
3. Consistency
4. Availability
5. Mobility
6. Location transparency
Andrew ES (AFS)

- build a scenario

- suggest (more than direct costing)

- personal comments (eh)
- No unit on site
- Has no infrastructure

Fire

Door

Dear

{[Diagram]}
dest = \text{port} \rightarrow \text{indirect address}

\text{dest} = \text{mod} \cdot \text{id} \mod N

\text{dest} = \text{pid} \times \text{x}

\text{Send} \left( \text{seq}, \text{dest} \right)

\text{Send} / \text{source}

\text{API}
if part a = part b then done = done

Source (part a, dock a) new (part b, dock b)

Receive (part a, dock a)

description