MANCHESTER 1824
The University of Manchester

Black-Box Cooperative Verification for Concurrent Programs

Fatimah Aljaafari
Supervisor: Dr. Lucas Cordeiro
Co-Supervisor: Dr. Mustafa A. Mustafa

University of Manchester
Department of Computer Science

MANCHESTER 1824
The University of Manchester

Motivation

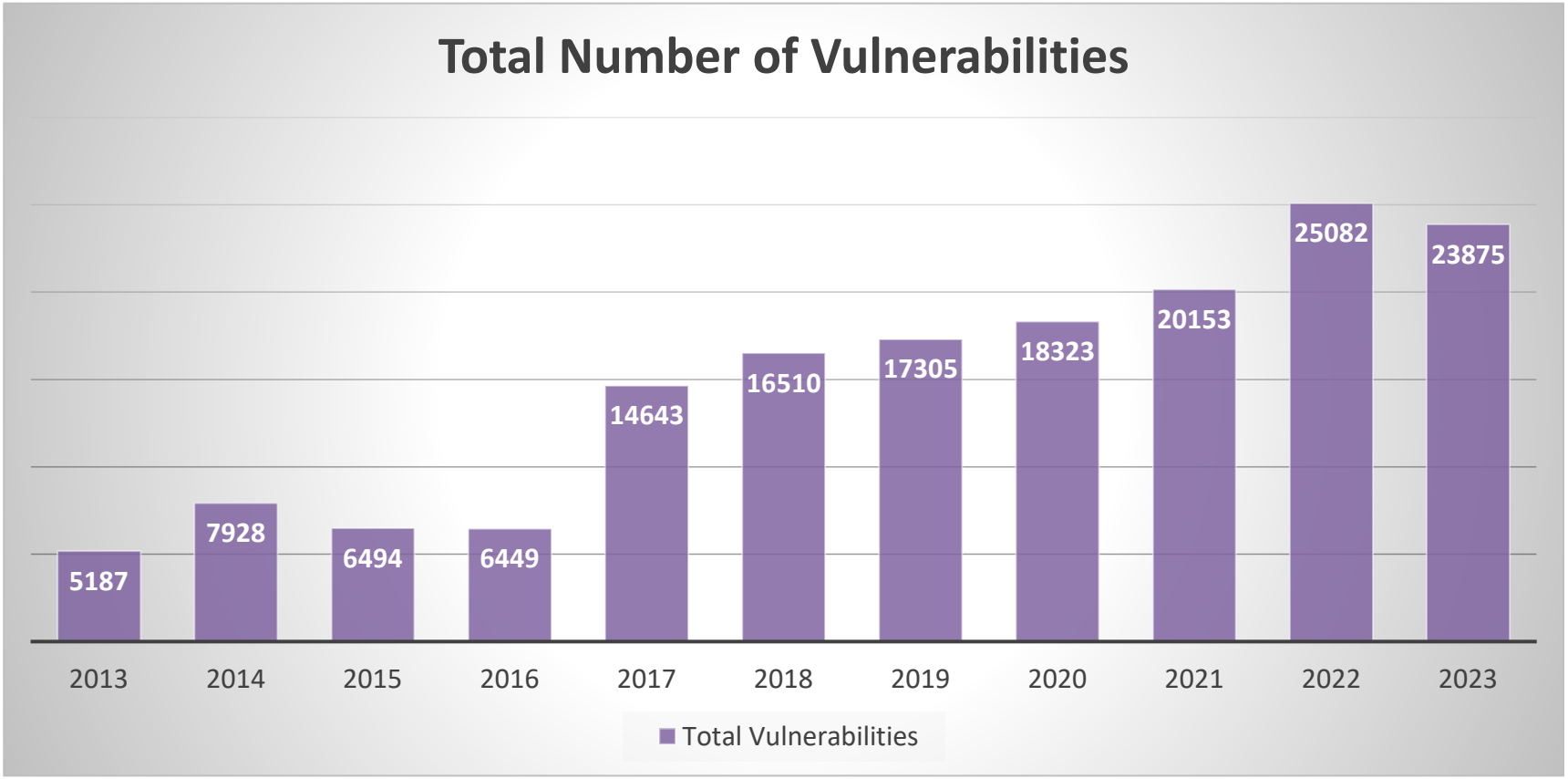


Figure 1:Total number of vulnerabilities reported from 2013 to 2023. Information adapted from Common Vulnerabilities and Exposures, url= https://www.cvedetails.com

Real-World Example

CVE-2014-0160:

In the OpenSSL cryptographic library, allowing the attacker to read the memory and obtain sensitive information from the affected system.

1st September 2023:

Manchester Airport have been warned they could face delays and cancellation after UK airspace was affected by a technical fault. The incident occurred due to an abnormality that forced the system to stop processing flight plans. The system was closed to maintain safety and required manual operation to continue service.

https://www.manchestereveningnews.co.uk/news/greater-manchester-news/every-cancelled-delayed-flight-manchester-27631376

MANCHESTER 1824
The University of Manchester

Research Problem

- Concurrency is becoming increasingly widespread in present-day software systems.

- Concurrent programs are relatively **complex**, posing unique challenges compared to sequential programs.

- Using automated testing and verification:
  - Fuzzing: is an automated testing technique that generates random inputs and checks whether an application crashes.
  - BMC: is a formal technique that proves the existence or absence of a path up to bound k.

MANCHESTER 1824
The University of Manchester

Research Problem

Why BMC and Fuzzing?

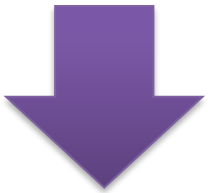
Dissimilar techniques.
BMC is a static analysis, and Fuzzing is an exploratory dynamic analysis.

	BMC	Fuzzing
Advantages	• Generate specific input	• Easy to use • The behavior is more authentic • Do not need an OM
Disadvantages	• State-space explosion • Over/under approximation	• Finding specific input • Not thread-aware

MANCHESTER 1824
The University of Manchester

Research Question

Given the current knowledge in software verification, no techniques harness BMC and fuzzing to verify concurrent programs.



The question of whether combining BMC and fuzzing improves bug finding in concurrent programs remains open.

MANCHESTER 1824
The University of Manchester

Objectives:

Introduce EBF – Black-Box Cooperative Verification for Concurrent Programs

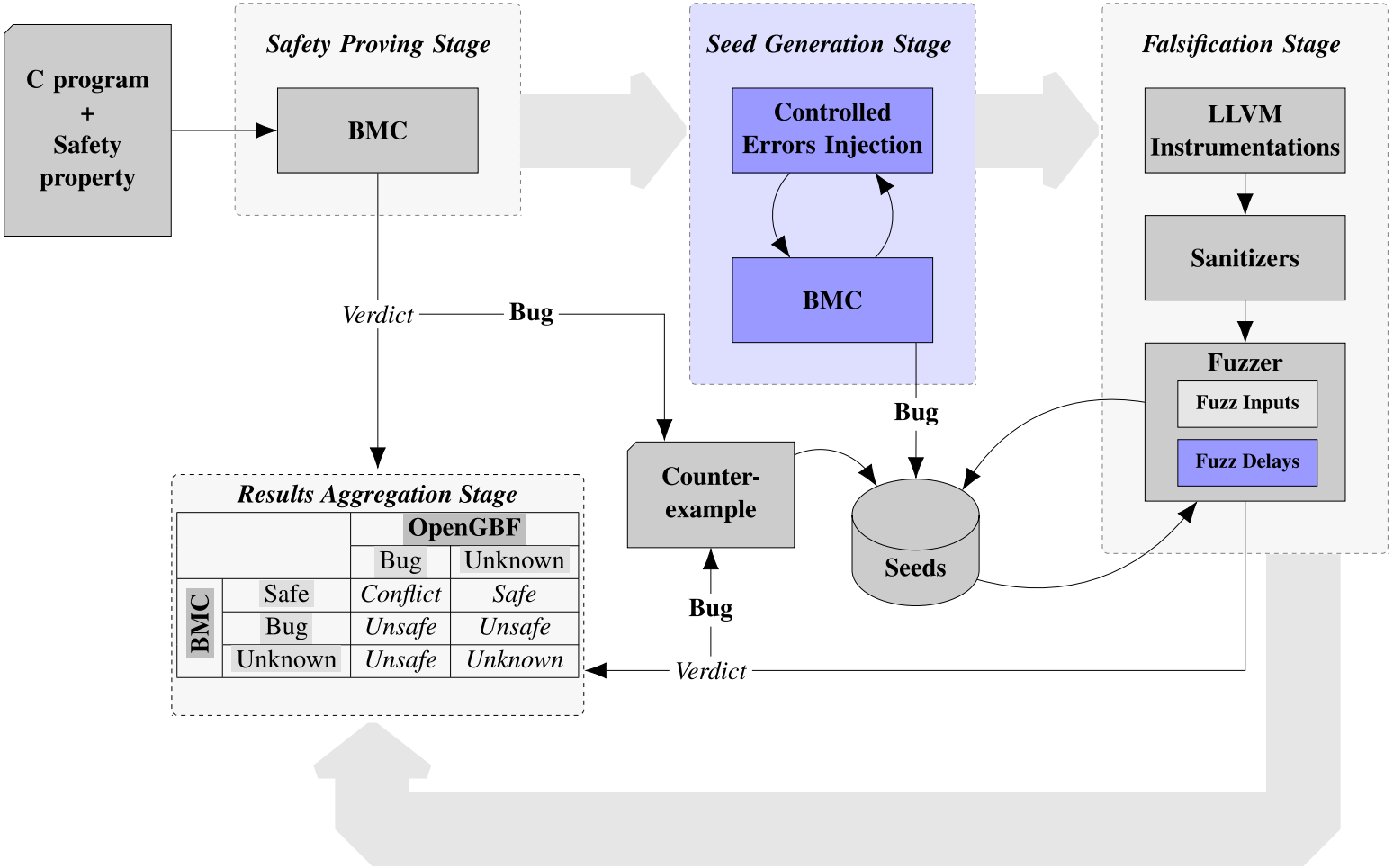
Propose open-source state-of-the-art concurrency-aware gray-box fuzzer

Combines the strengths of BMC with the flexibility of our concurrency-aware gray-box fuzzer.

Incorporates a result decision matrix

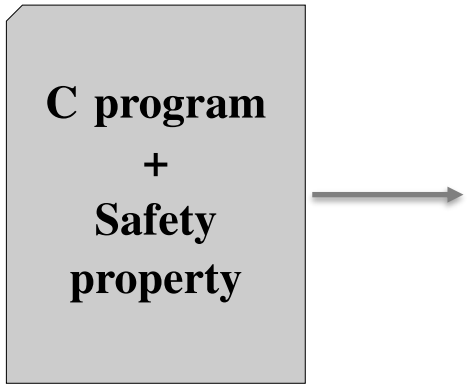
MANCHESTER 1824
The University of Manchester

EBF Design



MANCHESTER 1824
The University of Manchester

EBF Design

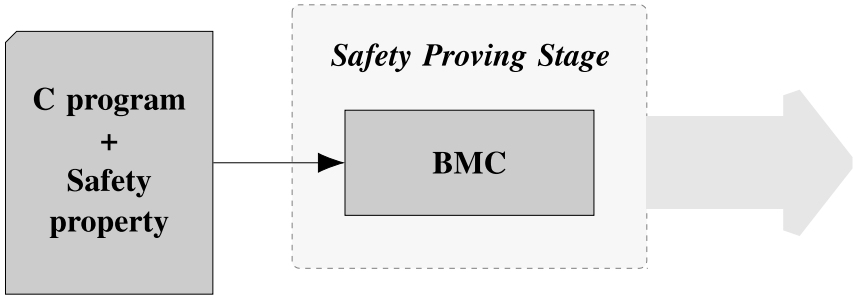



```
void reach_error() { assert(0); }
extern int __VERIFIER_nondet_int();

void foo(void * arg){
int a = __VERIFIER_nondet_int();
int b = __VERIFIER_nondet_int();
if((a + b) == 42) reach_error();
}
int main() {
pthread_t t1, t2;
pthread_create(&t1, 0, foo, 0);
pthread_create(&t2, 0, foo, 0);
pthread_join(t1, 0);
pthread_join(t2, 0);
}
```

→ Safety Properity

MANCHESTER 1824
The University of Manchester

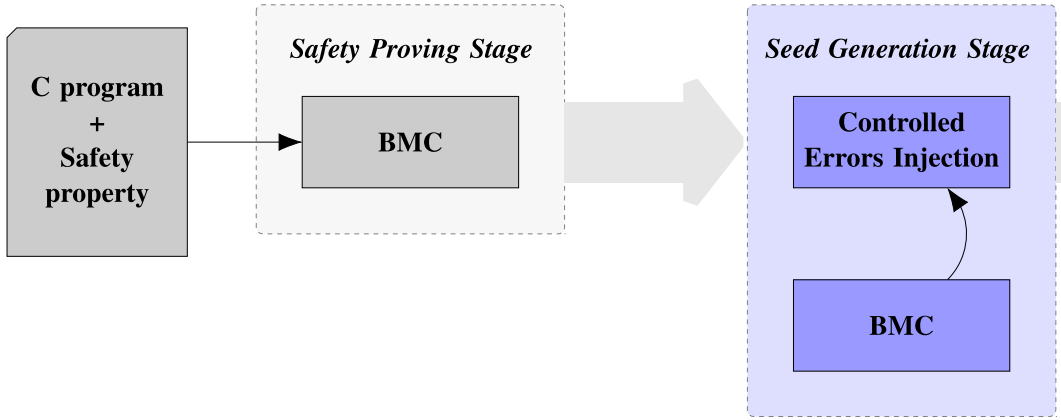
EBF Design




```
Counterexample:

State 5 file test.c line 9 function foo thread 1
-------------------------------------------------
  a = 42 (00000000 00000000 00000000 00101010)

State 6 file test.c line 10 function foo thread 1
-------------------------------------------------
  b = 0 (00000000 00000000 00000000 00000000)

State 7 file test.c line 5 function reach_error thread 1
-------------------------------------------------
Violated property:
  file test.c line 5 function reach_error
  assertion 0
  0


VERIFICATION FAILED
```

MANCHESTER 1824
The University of Manchester

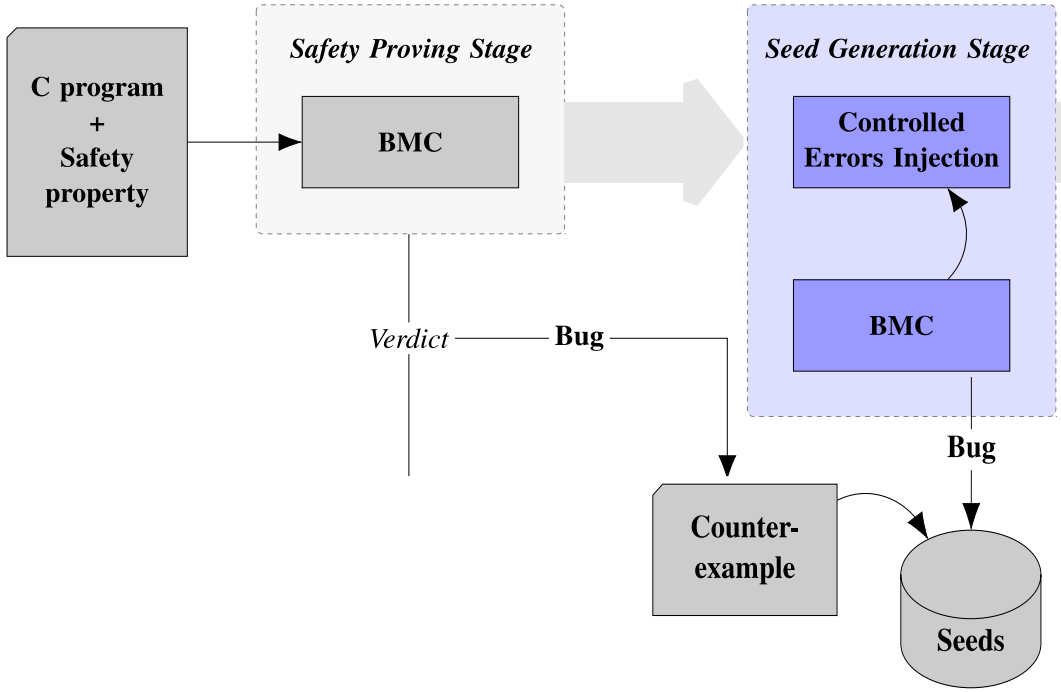
EBF Design




```
void reach_error() { assert(0); }
extern int __VERIFIER_nondet_in
void foo(void * arg){
int a = __VERIFIER_nondet_int();
int b = __VERIFIER_nondet_int();
if((a + b) == 42)
LABLE_1:;
reach_error();
} LABLE_2:;
int main() {
pthread_t t1, t2;
pthread_create(&t1, 0, foo, 0);
pthread_create(&t2, 0, foo, 0);
pthread_join(t1, 0);
pthread_join(t2, 0);
LABLE_3:;
}
```

MANCHESTER 1824
The University of Manchester

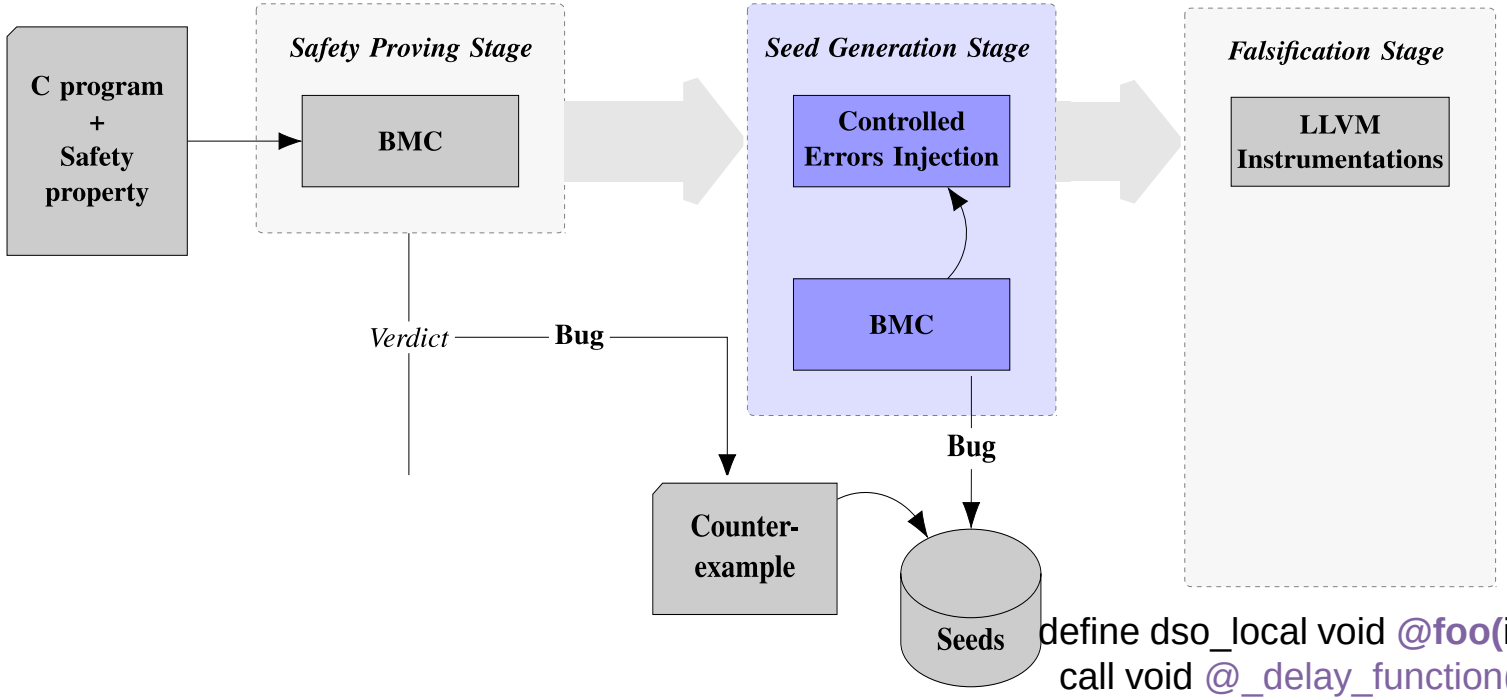
EBF Design




```
void reach_error() { assert(0); }
extern int __VERIFIER_nondet_in
void foo(void * arg){
int a = __VERIFIER_nondet_int();
int b = __VERIFIER_nondet_int()
if((a + b) == 42)
LABLE_1::;
reach_error();
} LABLE_2::;
int main() {
pthread_t t1, t2;
pthread_create(&t1, 0, foo, 0);
pthread_create(&t2, 0, foo, 0);
pthread_join(t1, 0);
pthread_join(t2, 0);
reach_error();
}
```

MANCHESTER 1824
The University of Manchester

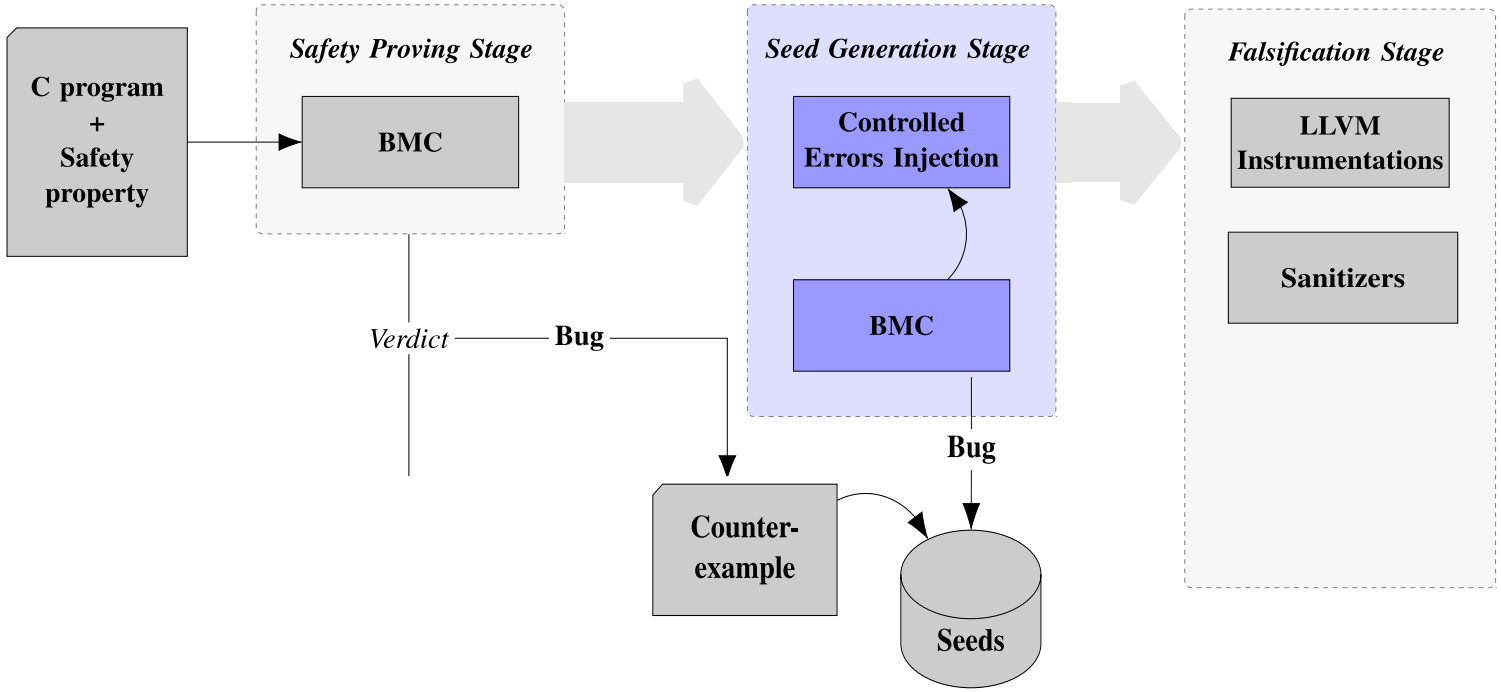
EBF Design




```
define dso_local void @foo(i8* %0) #0 {
  call void @_delay_function()
  %2 = alloca i8*, align 8
  call void @_delay_function()
  %bitcast_ebf = bitcast i8** %2 to i8*
  call void @_delay_function()
  call void @EBF_alloca(i8* getelementptr
inbounds
```

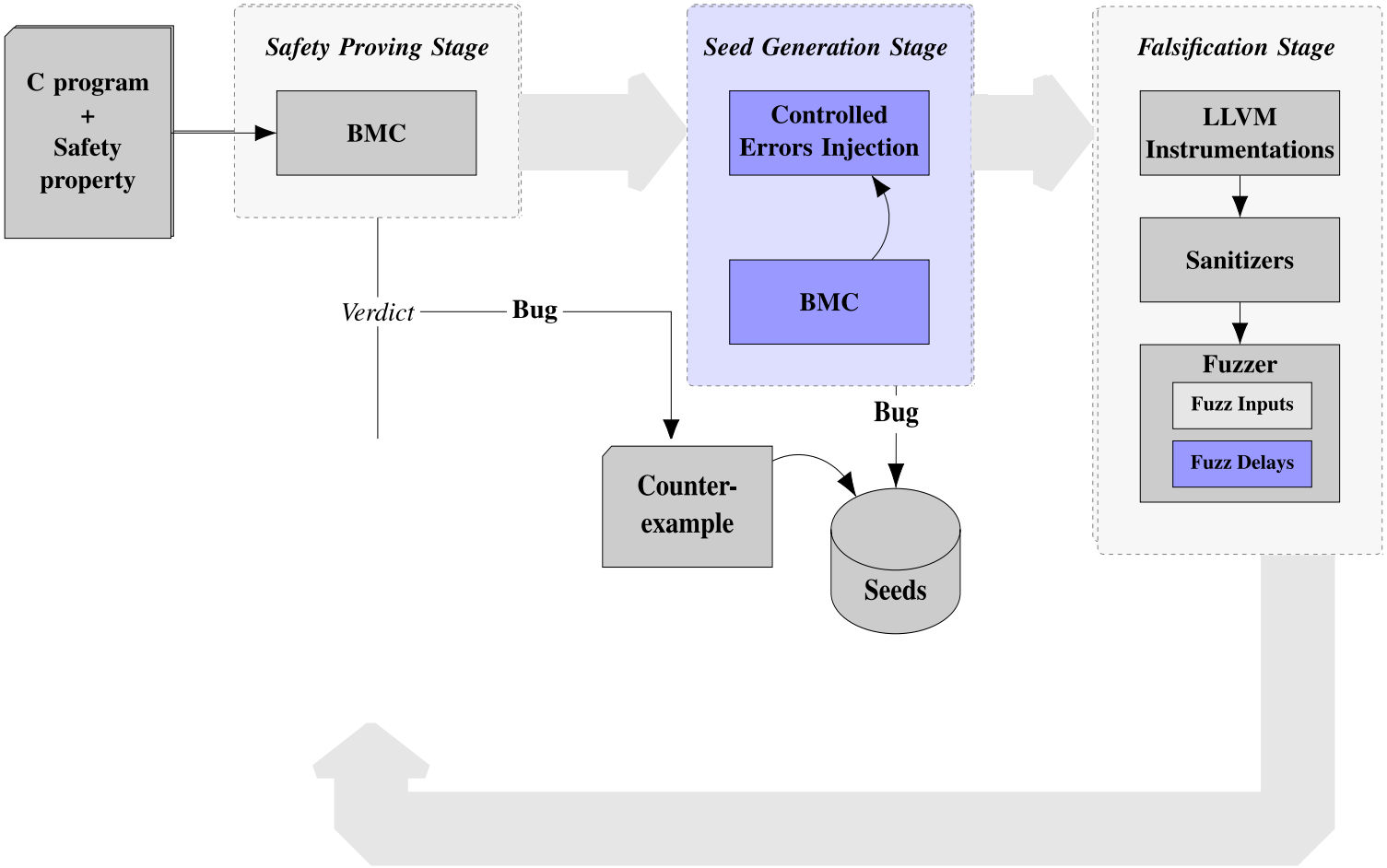
MANCHESTER 1824
The University of Manchester

EBF Design



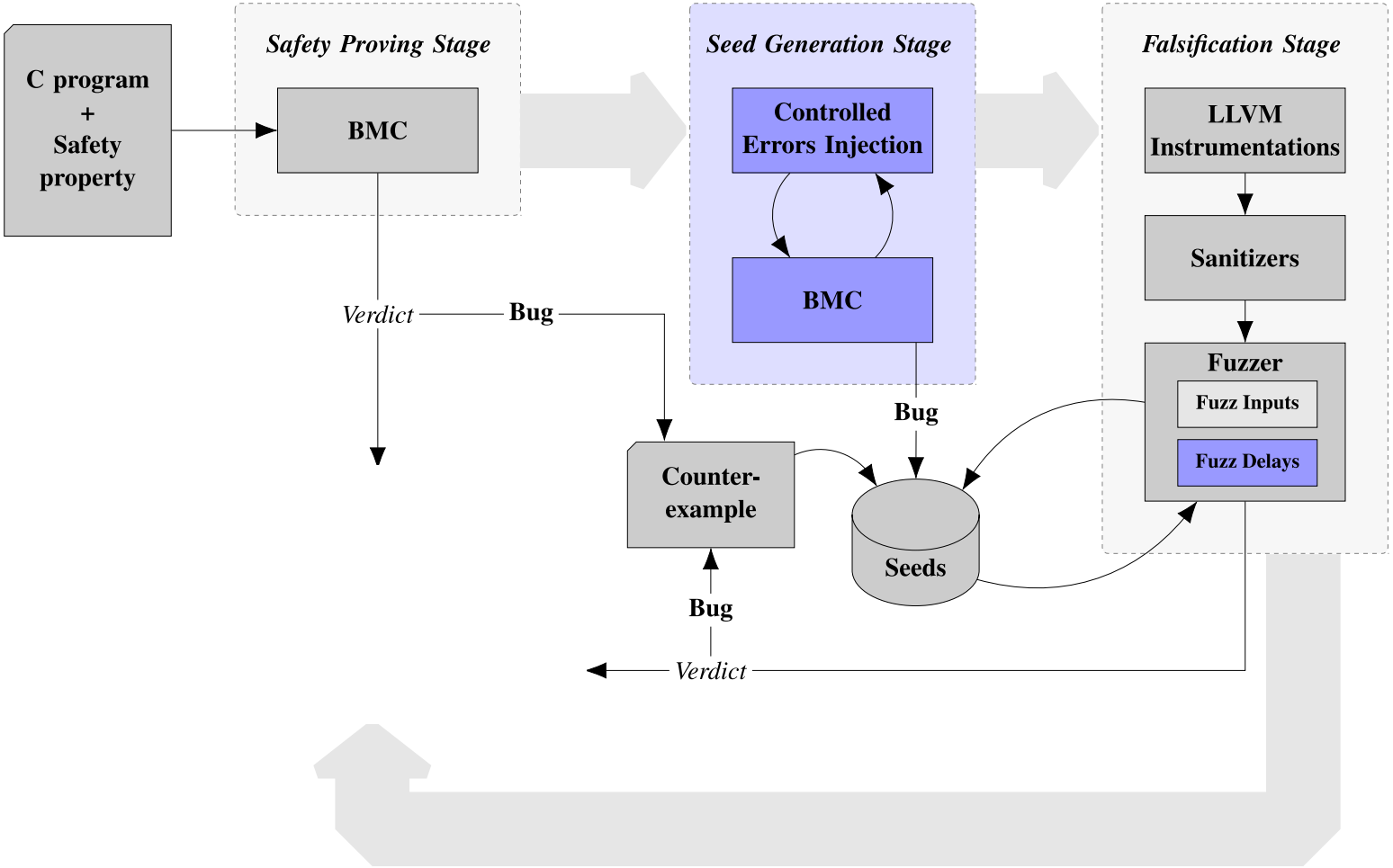
MANCHESTER 1824
The University of Manchester

EBF Design



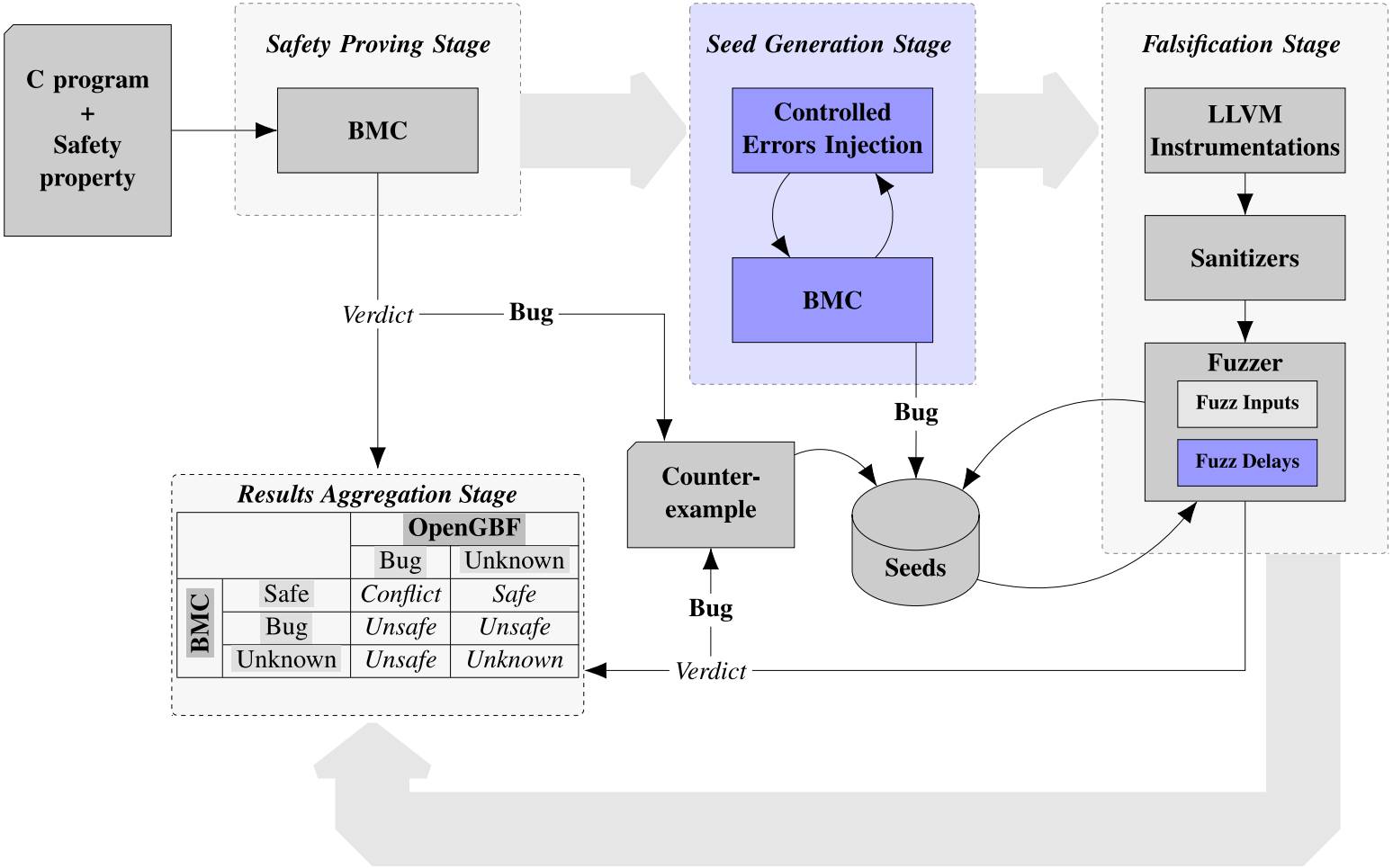
MANCHESTER 1824
The University of Manchester

EBF Design



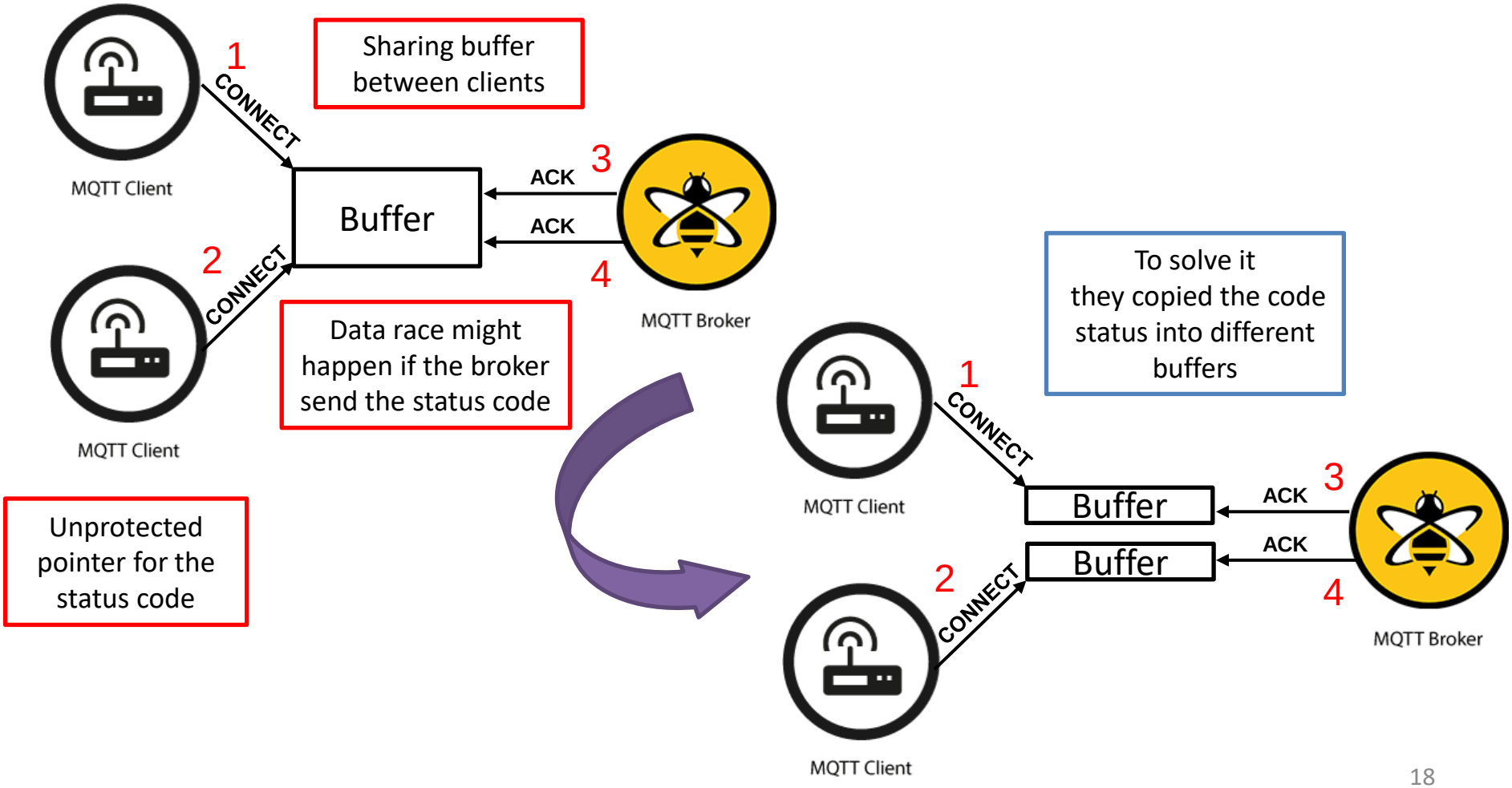
MANCHESTER 1824
The University of Manchester

EBF Design



WolfMQTT Verification

wolfMQTT library is a client implementation of the MQTT protocol written in C for **IoT devices**.



WolfMQTT Verification

Fixes for multi-threading issues #209

Merged **embhorn** merged 1 commit into `wolfSSL:master` from `dgarske:mt_suback` on 3 Jun 2021

Conversation 2 | Commits 1 | Checks 0 | Files changed 4

dgarske commented on 2 Jun 2021 — Contributor

1. The client lock is needed earlier to protect the "reset the packet state".
2. The subscribe ack was using an unprotected pointer to response code list. Now it makes a copy of those codes.
3. Add protection to multi-thread example "stop" variable.
   Thanks to Fatimah Aljaafari (**@fatimahkj**) for the report.
   ZD 12379 and PR **Data race at function MqttClient_WaitType** #198

`Fixes for three multi-thread issues: ...` ✕ `78370ed`

dgarske requested a review from **embhorn** 15 months ago

dgarske assigned **embhorn** on 2 Jun 2021

embhorn approved these changes on 3 Jun 2021 — View changes

MANCHESTER 1824
The University of Manchester

Any Question?