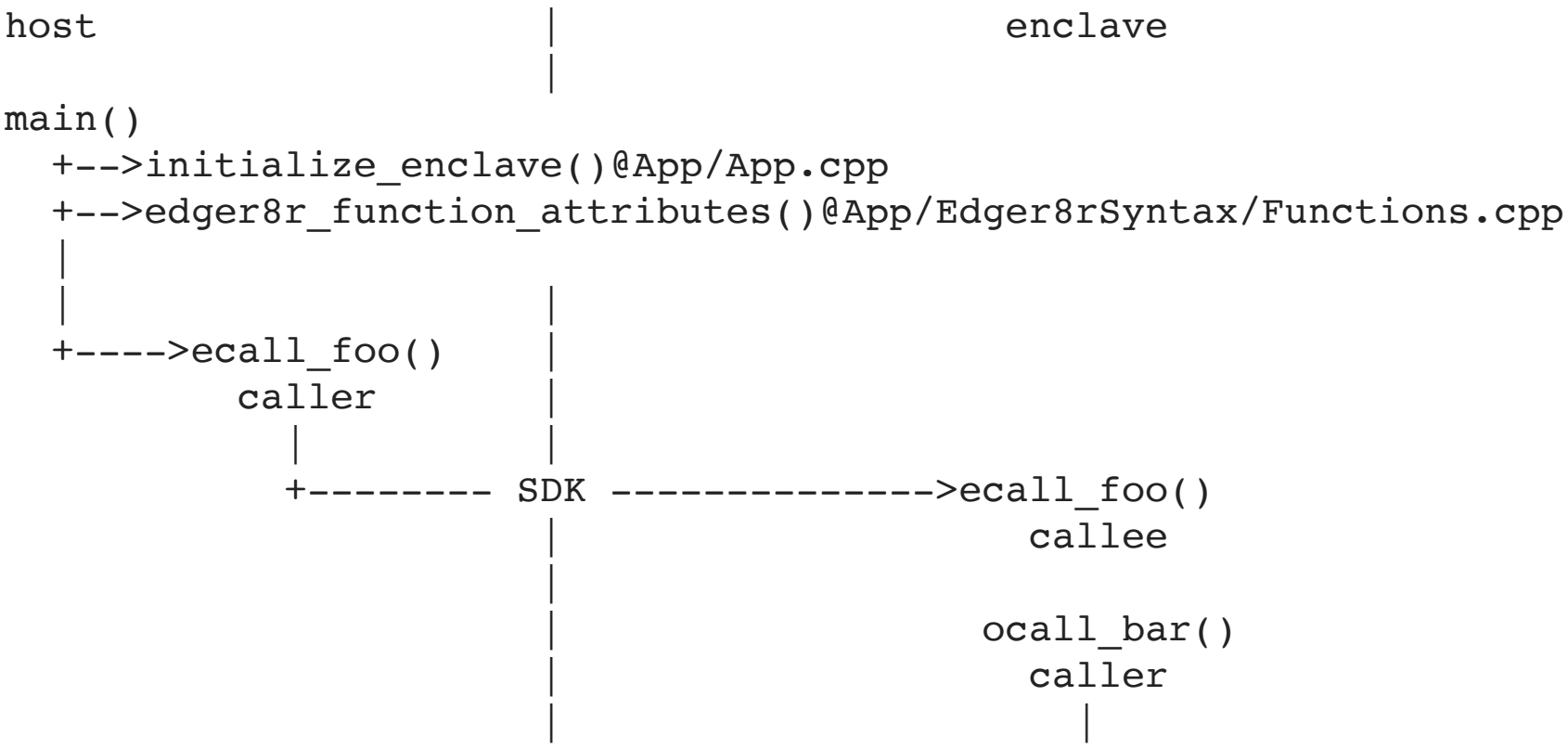
TUTORIAL ON SGX PROGRAMMING ON LINUX

YUZHE TANG

SEP. 29, 2017

SGX PROGRAMMING TUTORIAL

EXECUTION FLOW (OVERVIEW)



OCALL/ECALL

- ecall: caller outside enclave, and callee inside enclave
  - used to call into an enclave procedure
- ocall: caller inside enclave, and callee outside enclave
  - used to do systems service inside enclave

FUNCTION DEFINITION/DECLARATION

	ecall	**ocall**
invoke	App.cpp	Enclave.cpp
define	Enclave.cpp	App.cpp
declare	Libc.edl	Enclave.edl

- ecall:
  - invoke: `ecall_foo(global_eid,&retval,p)`@App.cp
  - define: `int ecall_foo(int* p)`@Enclave.cpp
  - declare: `int ecall_foo([in,count=l] int* p)`@Li
- ocall:
  - invoke: `ocall_bar(buf);`@Enclave.cpp
  - define: `void ocall_bar(const char *str)`@App.cp
  - declare:

    `void ocall_bar([in, string] const char *str`

ARGUMENT PASSING

- pass by pointer: SDK will deep-copy pointee to enclave
  - you need to annotate the argument pointer with its size of pointee
  - `int ecall_foo(int i, [in,count=1] int* p`
- pass by primitive: shallow copy