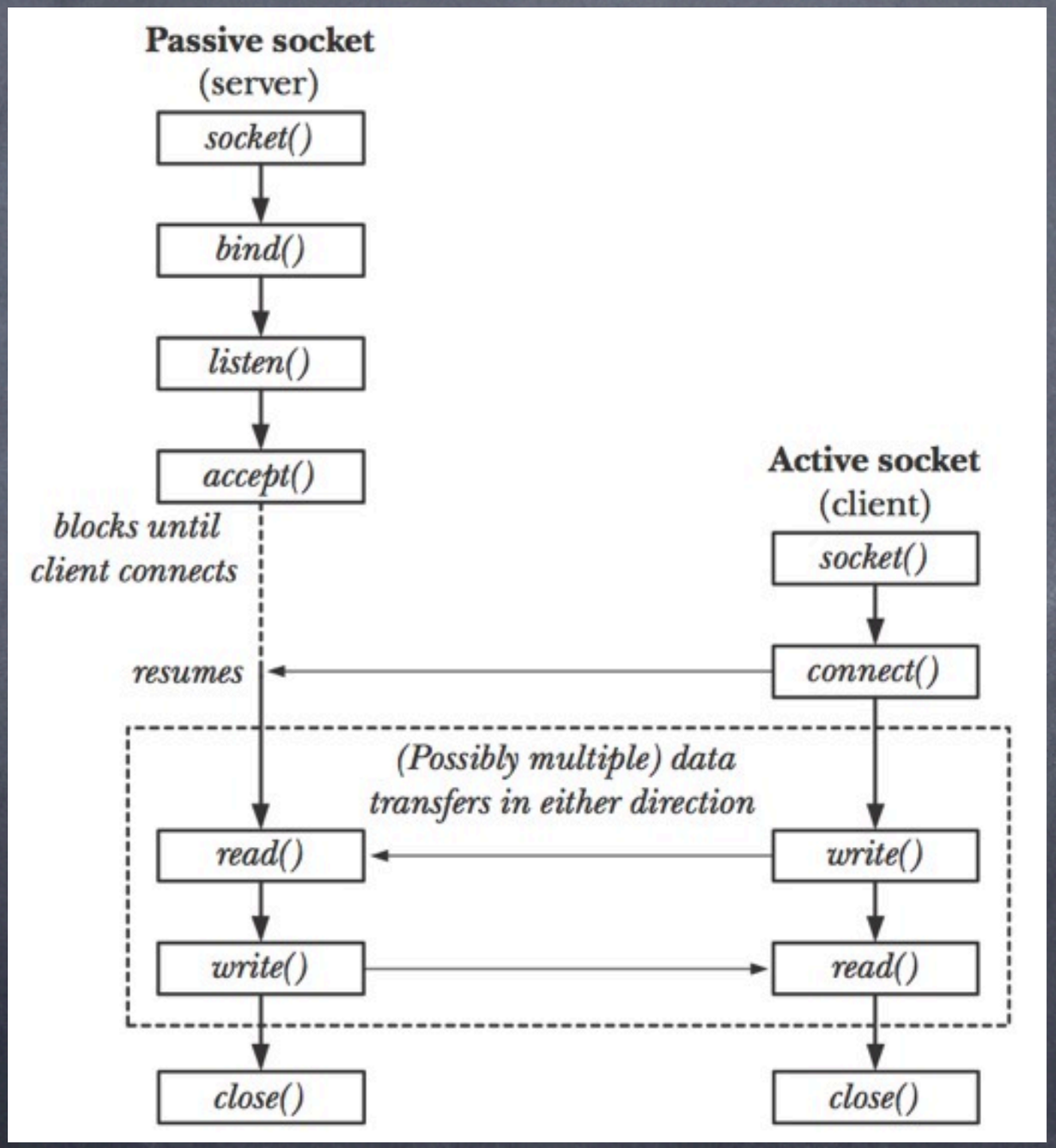
Sockets 101

Vaibhav Bajpai

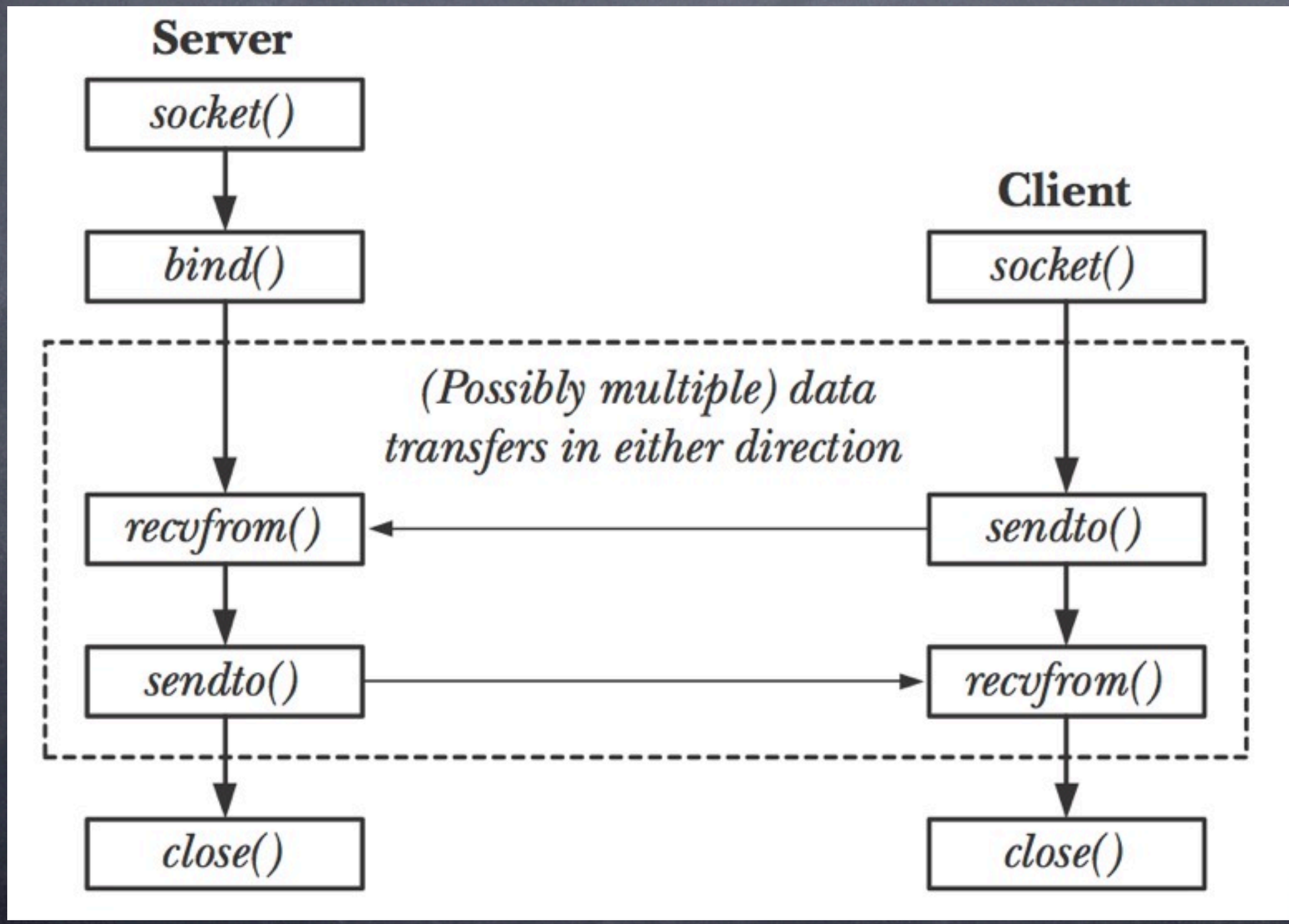
Introduction

- method of IPC allowing data exchange b/w applications
  - either on same host
  - or on different hosts connected by a network
- The first socket API came out in 1983!

Stream Sockets



Datagram Sockets



Creating a Socket

```
fsock = socket(domain, type, protocol)
```

- domain: AF_UNIX, AF_INET, AF_INET6, et al ...
- type: SOCK_STREAM, SOCK_DGRAM, et al ...

Generic Socket Addresses

```
bind(int fsock,
     const struct sockaddr * addr,
     socklen_t addrlen);
```

```
# include <sys/socket.h>

struct sockaddr {
  __uint8_t     sa_len;          /* total length */
  sa_family_t   sa_family;       /* address family */
  char          sa_data[...];    /* address value */
};
```

IPv4 Socket Address

```
#include <sys/socket.h>
#include <netinet/in.h>

struct sockaddr_in {
    __uint8_t         sin_len;          /* address length */
    sa_family_t       sin_family;       /* address family */
    in_port_t         sin_port;         /* transport layer port */
    struct in_addr    sin_addr;         /* IPv4 address */
};

struct in_addr {
    uint8_t           s_addr[4];
};
```

IPv6 Socket Address

```
#include <sys/socket.h>
#include <netinet/in.h>

struct sockaddr_in6 {
   __uint8_t       sin6_len;          /* address length*/
   sa_family_t     sin6_family;       /* address family */
   in_port_t       sin6_port;         /* transport layer port */
   __uint32_t      sin6_flowinfo;     /* flow information */
   struct in6_addr sin6_addr;         /* IPv6 address */
   __uint32_t      sin6_scope_id;     /* scope zone index */
};


struct in6_addr {
    uint8_t        s6_addr[16];
};
```

Mapping Names to Addresses

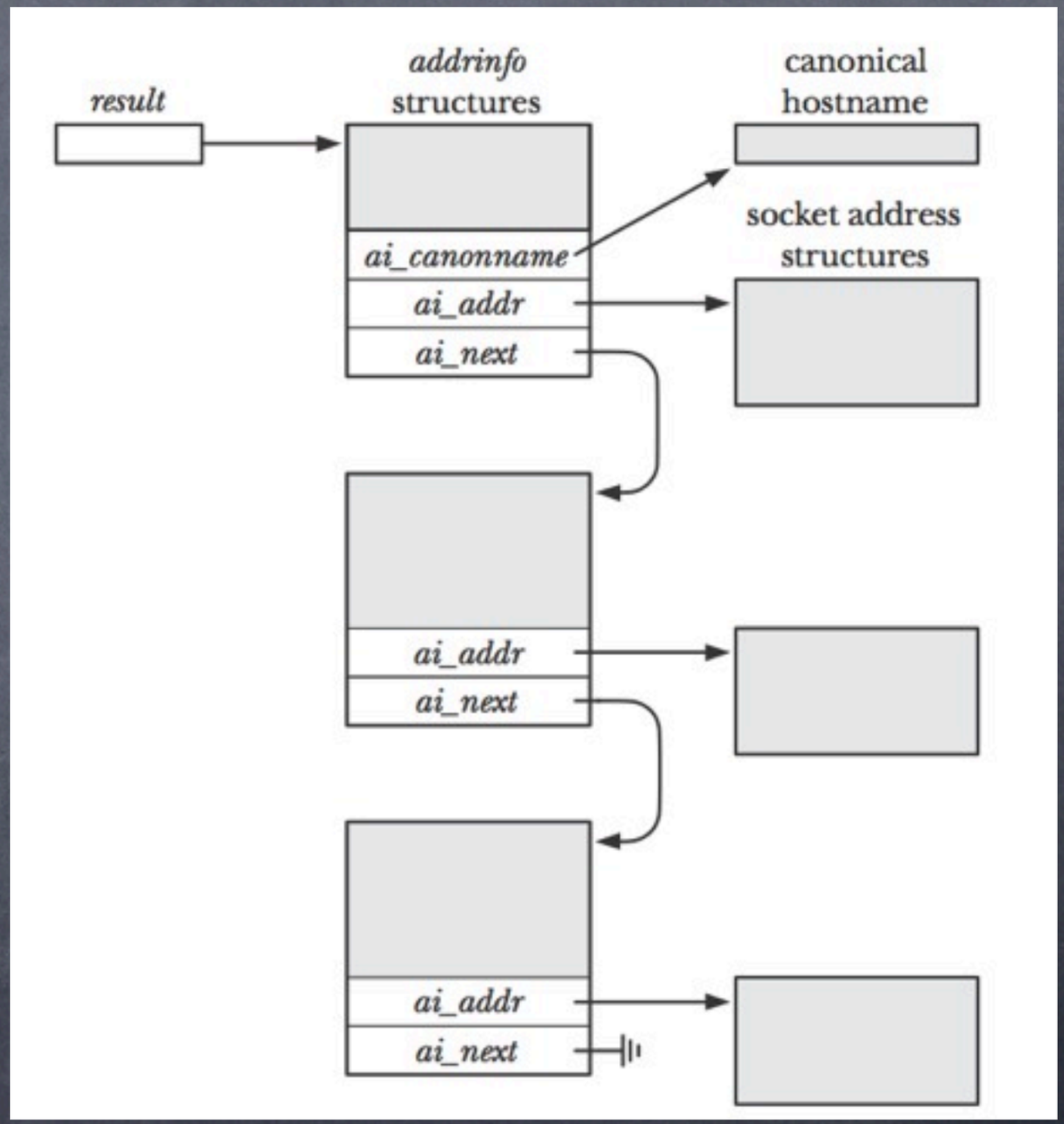
```
int getaddrinfo(const char *host,
                const char *service
                const struct addrinfo *hints,
                struct addrinfo **result);
```

```
void freeaddrinfo(struct addrinfo *);
```

Mapping Names to Addresses

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

struct addrinfo {
    int              ai_flags;
    int              ai_family;
    int              ai_socktype;
    int              ai_protocol;
    socklen_t        ai_addrlen;
    char            *ai_canonname;
    struct sockaddr  *ai_addr;
    struct addrinfo  *ai_next;
};
```



Event Driven Loops

```
select(int nfds,
       fd_set *readfds,
       fd_set *writefds,
       fd_set *exceptfds,
       struct timeval *timeout);
```

```
void FD_ZERO(fd_set *fdset);
void FD_SET(int fd, fd_set *fdset);
void FD_CLR(int fd, fd_set *fdset);
int FD_ISSET(int fd, fd_set *fdset);
```

References

The Linux Programming Interface by Michael Kerrisk, No Starch Press