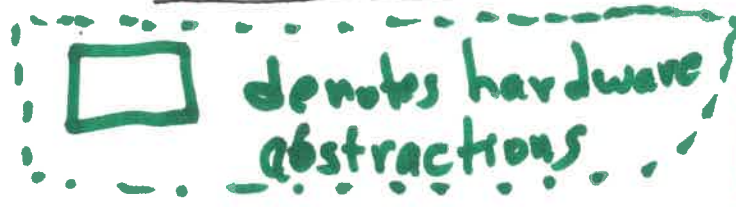
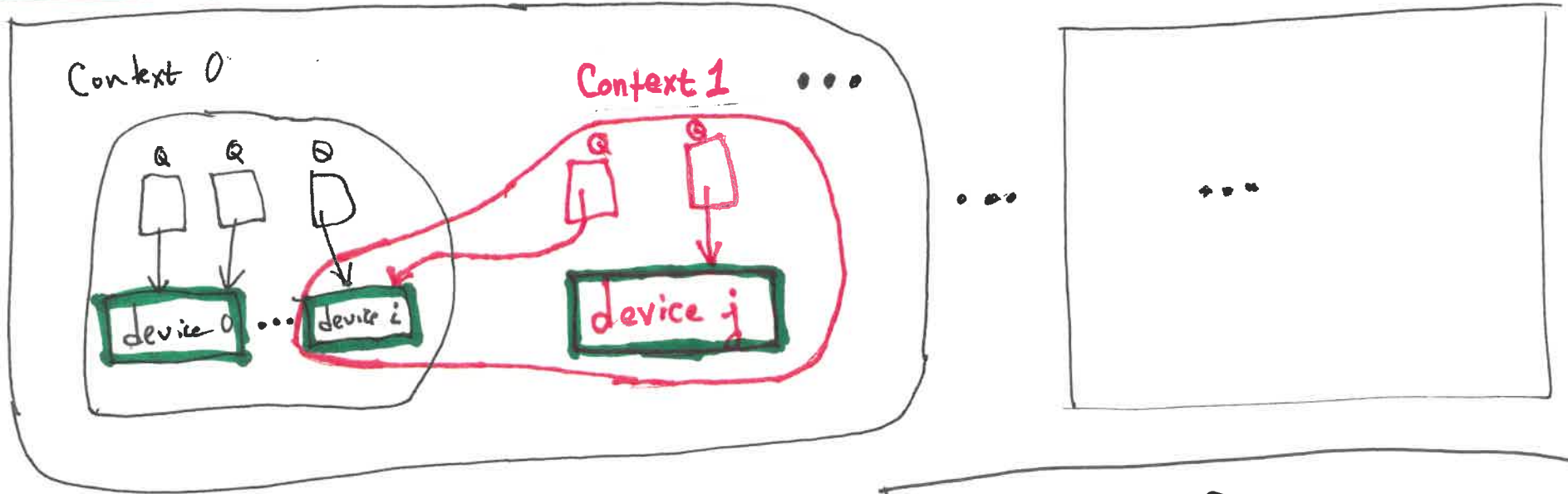


# OpenCL Platform Model

Platform 0



Platform k

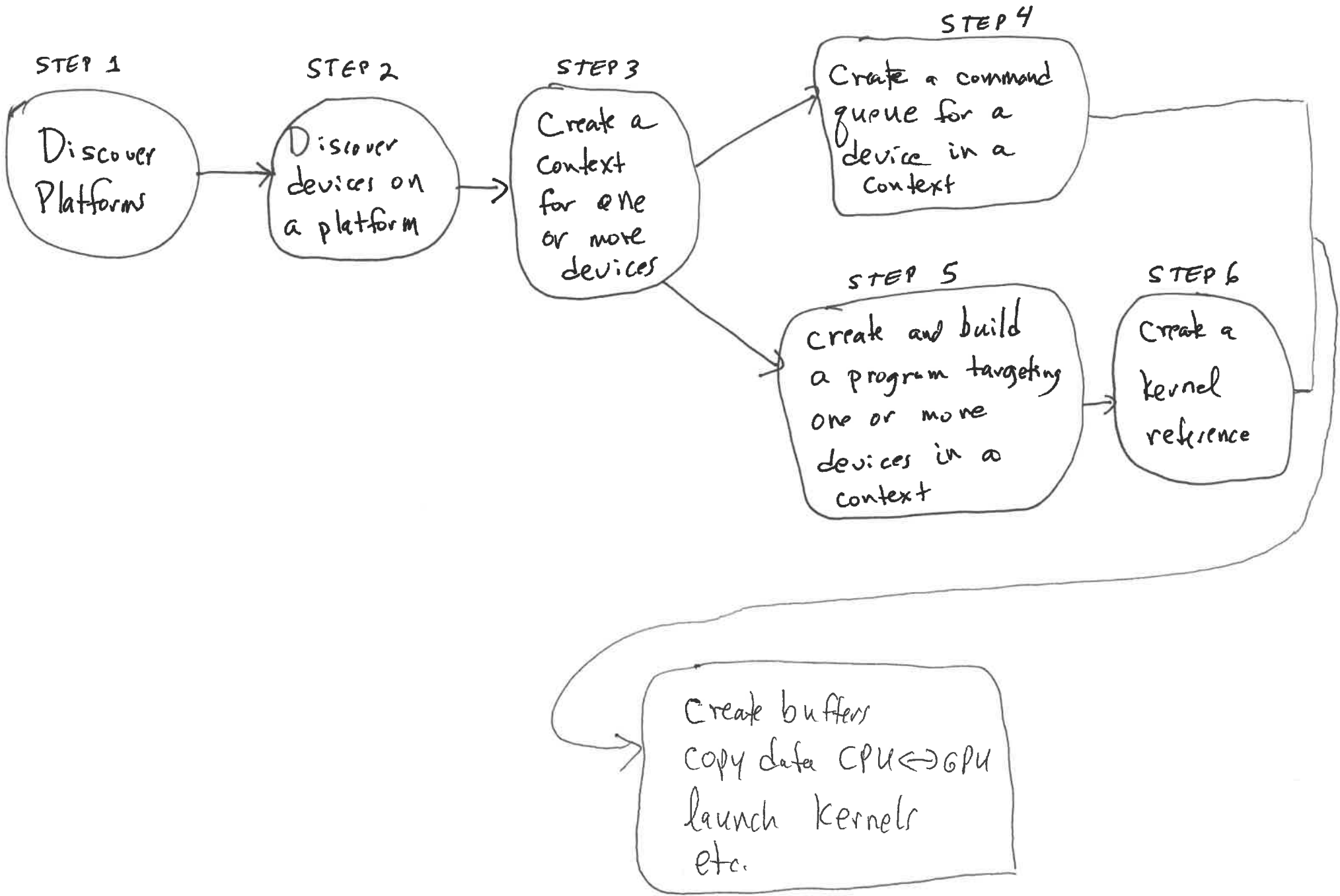


- Discover platforms & devices
- Create contexts & queues

Queues are for the device as a whole, NOT, for example, for individual SMs of a GPU.

- For most (if not all) of the course, we will implicitly assume:
  - 1 platform
  - 1 device
  - 1 context
  - 1 command queue

# Standard OpenCL Prolog



## OpenCL Startup Function prototypes

The functions for steps 1 and 2 are designed to use two calls: one to see how many results you will get, another to actually get them. (You can always assume a maximum size and hope.)

On return types: `clGet*` and `clBuild*` return error status code; `clCreate*` returns the object created (and error status is a pass-by-address parameter).

### STEP 1:

```
cl_int clGetPlatformIDs(cl_uint num_entries, cl_platform_id*
    platforms, cl_uint* num_platforms);
```

### STEP 2:

```
cl_int clGetDeviceIDs(cl_platform_id platform, cl_device_type
    deviceType, cl_uint numEntries, cl_device_id* devices,
    cl_uint* num_devices);
```

### STEP 3:

```
cl_context clCreateContext(const cl_context_properties* props,
    cl_uint numDevices, const cl_device_id* devices, CCErrCB
    notify, void* userData, cl_int* errorCode);
```

Final three parameters can be nullptr; the error callback, if used, has prototype:

```
void notify(const char* errorMessage,
    const void* pvtInfo, size_t pvtInfoSize, // binary data
    void* user_data)
```

### STEP 4:

```
cl_command_queue clCreateCommandQueue(cl_context context,
    cl_device_id device, cl_command_queue_properties props,
    cl_int* errorCode);
```

### STEP 5:

```
cl_program clCreateProgramWithSource(cl_context context, cl_uint
    count, const char** strings, const size_t* lengths, cl_int*
    errorCode);
```

```
cl_int clBuildProgram(cl_program program, cl_uint numDevices,
    const cl_device_id *deviceList, const char* options,
    BPErrCB notify, void* userData);
```

Final three parameters can be nullptr; the error callback, if used, has prototype:

```
void notify(cl_program program, void* user_data)
```