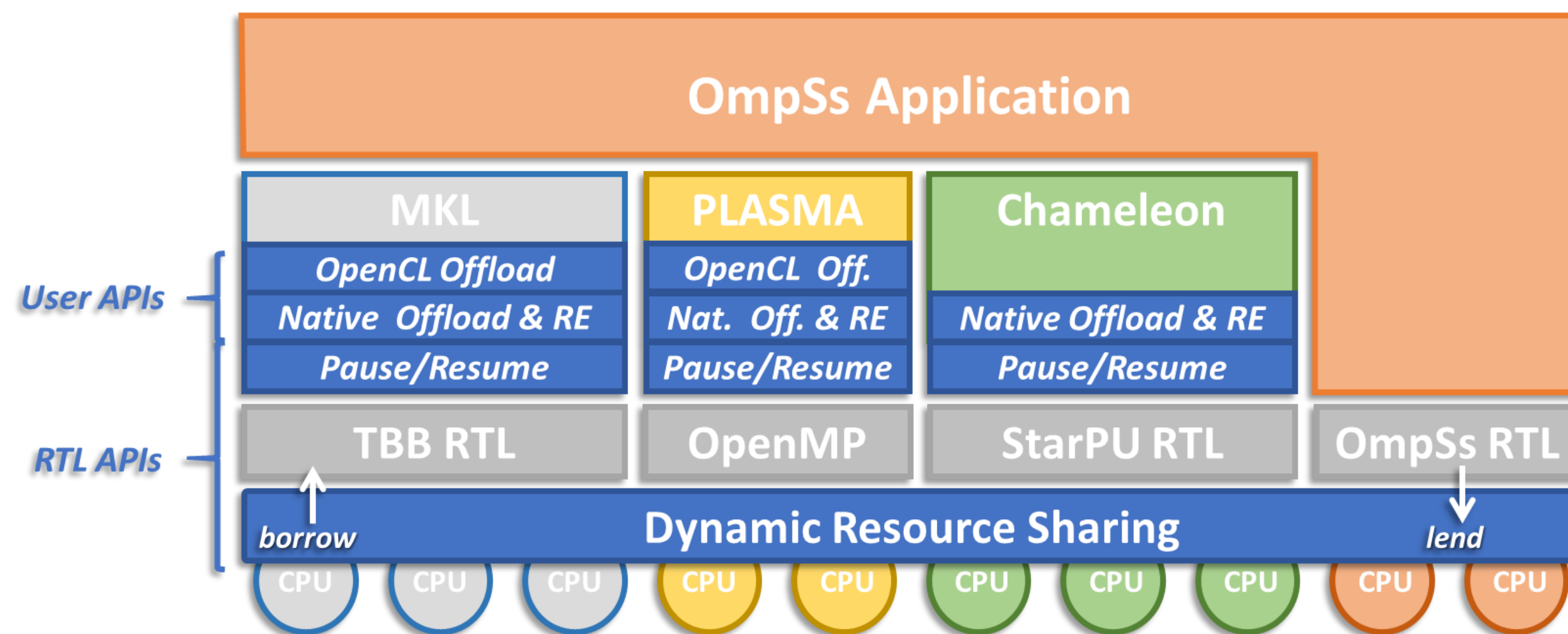


# INTERTWinE: Resource Manager APIs

## Resource Manager General Overview

The **Resource Manager** is designed to improve the interoperability and composability between parallel runtime systems and parallel libraries. To that end, the Resource Manager introduces four APIs to arbitrate access to CPUs between multiple parallel runtime systems and to improve the interoperability between task-based programming models and communication APIs. These four APIs are designed as low-level mechanisms that will be used to implement and enforce high-level user-defined policies.



## Offloading and Resource Enforcement APIs

The **Offloading** and **Resource Enforcement** APIs enable the encapsulation and instantiation of **task-based parallel kernels** from any sequential or parallel **application** (pthreads, OpenMP, Java, etc.). Offload may work at two different layers: OpenCL (optional) and Native (mandatory). Resource Enforcement allows resource ownership to be established once the application or library is initialized.

### → Native Offload & Resource Enforcement:

```
void tbrs_spawn_kernel_on_cpus(void *data, void(*function)(void *),
void *args, cpu_set_t *cpu_set);
```

### → OpenCL Offload & Resource Enforcement:

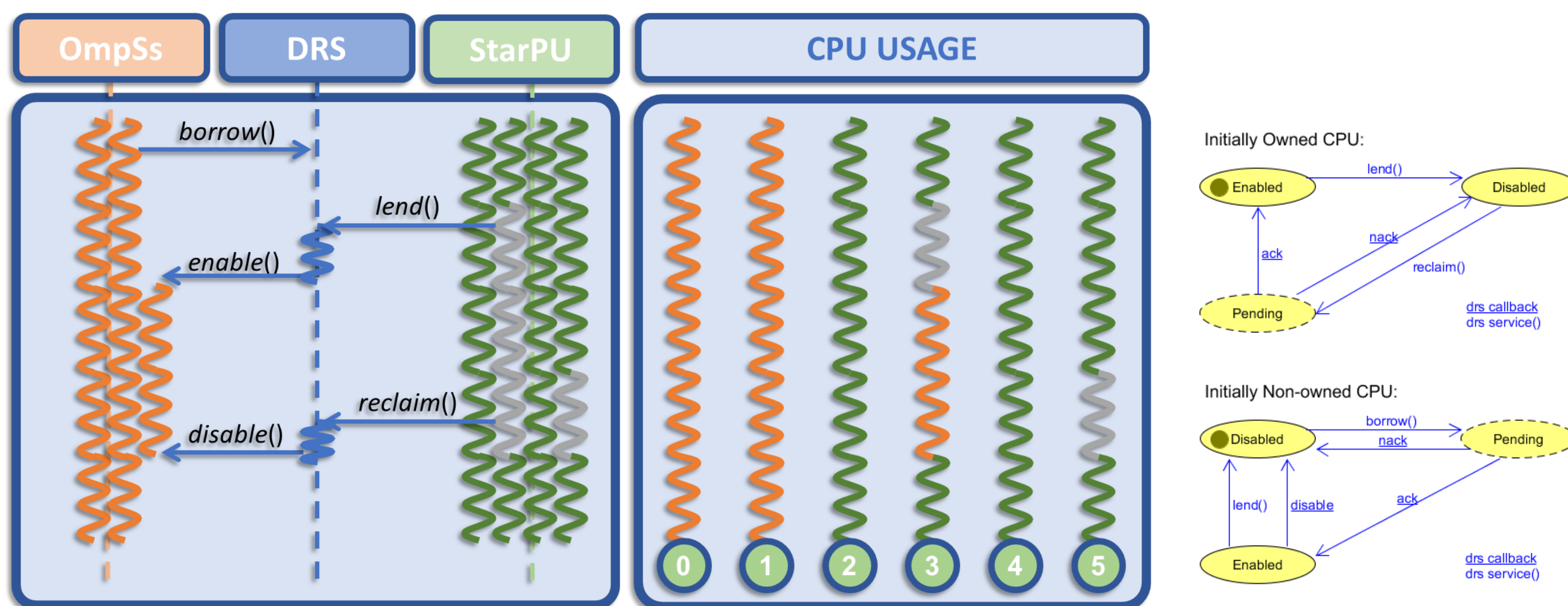
```
subdevice = clCreateSubDevices(device, props, num_dev, subdevices, 0);
ctx = clCreateContext(0, 1, &subdevice, 0, 0, 0);
queue = clCreateCommandQueueWithProperties(ctx, dev[0], 0, NULL);

[...]

clEnqueueNativeKernelWithType(queue, &libMATH_compute_kernel, &args,
sizeof(args), 0, NULL, NULL, CL_NATIVE_KERNEL_OMPSS, 0, NULL, &h->event);
```

## Dynamic Resource Sharing API

Sometimes the programmer misestimates the resources required to execute a parallel kernel which may result in either oversubscription or undersubscription. **Dynamic Resource Sharing** allows **task-based runtime systems** to **lend** owned resources when they are unused and to **borrow** non-owned resources when its owner is not using them.



## Task Pause and Resume API

The explicit **Task Pause and Resume API** is a set of services that allows third party **libraries** (or other software elements) to interact with the scheduler of a **task-based runtime system**. The program may inform the scheduler that a task is going to block, so that other tasks can be executed until the blocked task becomes ready again.

