```
      do {
            /* Get a handle to remote memory for software cache-access */
            sw_cache_hndl =
mrapi_rmem_get(AGREED_ID_FOR_SW_CACHE,MRAPI_ACCESS_TYPE_SW_CACHE,
                                                                  &status);
            // CHECK STATUS FOR ERROR
            if (status != MRAPI_SUCCESS) {
                  ERR("Unable to get remote memory for sw cache");
            }

      } while (!get_successful(&status));

      /* Use the handle to attach to the remote memory */
      mrapi_rmem_attach(sw_cache_hndl,
                        MRAPI_ACCESS_TYPE_SW_CACHE,
                        &status);

      // CHECK STATUS FOR ERROR
      if (status != MRAPI_SUCCESS) {
        ERR("Unable to attach to remote memory for sw cache");
      }

      do {
            /* Get a handle to remote memory for DMA-access */
            dma_hndl =
mrapi_rmem_get(AGREED_ID_FOR_DMA,MRAPI_ACCESS_TYPE_DMA,&status);

            // CHECK STATUS FOR ERROR
            if (status != MRAPI_SUCCESS) {
                  ERR("Unable to get remote memory for DMA");
            }

      } while (!get_successful(&status));

      /* Use the handle to attach to the remote memory */
      mrapi_rmem_attach(dma_hndl,MRAPI_ACCESS_TYPE_DMA,&status);

      // CHECK STATUS FOR ERROR
      if (status != MRAPI_SUCCESS) {
        ERR("Unable to attach to remote memory for DMA");
      }

      unsigned int num_entities_processed = 0;

      // Pair of request objects to allow us to wait for DMA operations on
      // either of two result buffers
      mrapi_request_t[2] request;

      Entity next_entity_to_process;

      bool first = true;
```