6 FAQ

Q: Is a reference implementation available? What is the intended purpose of the reference implementation?

A: An example implementation is available for download at the Multicore Association website. The example implementation models the functionality of the specification and does not intend to be a high performance implementation.

Q: Does MCAPI provide binary interoperability?

A: MCAPI provides a set of communications primitives that supports software source code portability. Interoperability is not addressed at this stage and is left to the respective MCAPI implementations. Implementers should however be encouraged to ensure implementation interoperability, as much as possible. We may consider an interoperable messaging protocol in future MCAPI versions.

Q: What is the rationale behind MCAPI's use of unidirectional channels as opposed to bidirectional channels?

A: MCAPI is meant to serve as an extremely lightweight and thin communications API for use by applications and by other messaging systems (such as a light-weight sockets) layered on top of it. MCAPI is expected to run on both multicore processors running real operating systems and running extremely thin run-time environments in bare-metal mode. MCAPI is also meant largely for on-chip multicore environments where communications is reliable.

MCAPI chose to go with unidirectional channels because they use less resources on both the send and receive sides. A bidirectional channel would need to consume twice the resources on both the send and receive sides (for example, hardware queues).

Many of our use cases require only unidirectional channels. Although unidirectional channels could be implemented using a bidirectional channel, it would waste half the resources. Conversely, bidirectional channels can be implemented on top of unidirectional channels when desired. Locking mechanisms on the send and receive sides provided by the underlying system on top of which MCAPI is built (or provided by a layer such as MRAPI) can be further used to achieve atomicity between sends and receives on each side.

Bidirectional channels are particularly effective in error-prone environments. MCAPI's target is multicore processors, where the communications is expected to be reliable.

Q: How does MCAPI compare to other existing communication protocols for multicore?

A: MCAPI is a low level communication protocol specifically targeting statically configured heterogeneous multicore processors. The static property allows implementations to have lower overhead than other communication protocols such a sockets or MPI.

Q: Regarding message sends, when can the sender reuse the buffer?

A: With blocking calls, the send can reuse once the call has returned. With non-blocking calls, it is the responsibility of the sender to verify using API calls that the buffer can be reused.

Q: Can MCAPI support a software-controlled implementation of global power management? If so, how?

A: MCAPI provides communication functionality between different cores in a multicore multiprocessor and can function at user level and system level. If the global power management scheme requires data to be passed between different cores, an MCAPI implementation could be used to provide the communications functionality.