be a potential attacker. This admittedly is the case for various Digital Rights Management (DRM) or payment processing systems. For these systems the end user is considered a potential enemy, who might want to illegally make a copy of a movie, or clone credit card information. Indeed, only then the device would like to protect its *runtime* processing. Otherwise an attacker who managed to steal the device would not be able to get it to start doing the processing of sensitive data in its RAM, without providing a proper unlock password or key in the first place. It's worrying that the industry has been aggressively advertising various DRM-friendly technologies as protecting the user, while in fact they have an opposite effect, degrading trustworthiness of the user devices (from the user point of view, that is).

An exception would be a tamper-proof design which allowed for reliable read-only access for all the firmware (and preventing access only to key-holding storage), but it seems like existing devices (specifically microcontrollers) do not support such a mode today, at least the author is not aware of any.