Untrusted platforms: Software provider doesn't trust owner of hardware on which the application runs. The examples are as follows:

- Cloud Computing.
- Digital rights management.
- Proprietary software.
- Smart Grids.
- Online Gaming.
- E-voting

In general, there are at least two different needs the applications have:

⇒ Tamper proof. In some cases may be secrecy (example: cellphones)

Remote Attestation:

⇒ In other cases like online gaming we need remote attestation. That is proving to a remote party that you are running a specific program. (Example: Digital Rights Management, smart grids, e-voting).

Major weakness of remote attestation: Doesn't ensure the authenticate software is not comprised by something like buffer overflow, format string etc.
Secure attestation Boot Process: This is employed in the paper *Terra: A Virtual Machine-Based Platform for Trusted Computing*

Example application iTunes which communicates with the Apple store.

1) First thing the hardware loads will be the Terra code on top of the hardware.

2) Before executing terra the hardware executes a hash function on the file and gets a checksum.

3) Terra communicates with lower level and provides its public key $P_T$ and lower level replies with a signed certificate, that is $C_T = \text{Cert}(S_H, P_T || h)$

Note: What makes hardware trusted? it is made to be tamper proof.

4) Similarly there is a certificate for the application running on the VM.
   Certificate for the application (iTunes) $C_I = \text{Cert}(S_T, P_I || h_I)$.

5) There is also a certificate the hardware: $C_H (S_D, P_H)$.

Note: The cryptography is needed only for remote attestation.ie not essential for just running an application on a VM.

6) We need to send $C_I$, $C_T$ and $C_H$ for remote authentication. The apple store will have $h_t = H(Terra)$ and $h_I = H(iTunes)$. The apple store can then encrypt the music or other files to encrypt the data using a separate session key.

Note: Whenever we transfer data we could either encrypt it or sign it using public key cryptography. (eg voting scenario when the voting software needs to send to the district)
What about gaming data? You need to ensure the gaming data is not tampered with using a network proxy so you again sign it.

For the data coming from the server needs to be encrypted. Therefore we establish a secure channel using the keys. ie Any channel that provides bi directional communication that cannot be tampered with.

Who is this for?

Originally for someone like a movie producer. Now used in cloud computing.
What if iTunes is a very big application?

Booting might be very time consuming. So we could use demand paging. When iTunes asks for its certificate we don’t compute the hash but pick up a previously computed hash placed in a table associated with the app.