



#### PSA-defined standard metadata (in red)

IGPIM - `psa_ingress_parser_input_metadata_t`  
 IGIM - `psa_ingress_input_metadata_t`  
 IGOM - `psa_ingress_output_metadata_t`  
 EGPIM - `psa_egress_parser_input_metadata_t`  
 EGIM - `psa_egress_input_metadata_t`  
 EGOM - `psa_egress_output_metadata_t`  
 EGDIM - `psa_egress_deparser_input_metadata_t`

#### User-defined metadata (in green)

IGH - ingress headers  
 IGM - ingress metadata  
 EGH - egress headers  
 EGM - egress metadata  
 NM - normal metadata for unicast and multicast packets  
 RESUBM - resubmit metadata  
 RECIRCUM - recirculate metadata  
 CI2EM - ingress-to-egress clone packet metadata  
 CE2EM - egress-to-egress clone packet metadata

#### Notes on parameters with direction 'inout'

IGOM, EGOM - input for Ingress/Egress because all PSA implementation must implement a specified subset of the fields of this struct before Ingress/Egress processing begins, e.g. `drop=false`.

IGH, EGH, IGM, EGM are inout for Ingress and Egress because we want parsers to be able to assign values to them before Ingress/Egress start, and we want the modified values to pass through to the deparsers.

IGH, EGH - inout for the deparsers, not because the data goes anywhere else after the deparser is complete, but because direction 'in' in P4\_16 prevents you from modifying a parameter inside of the control, and we wanted to permit the deparser code to update header checksums, e.g. for IPv4 headers, inside of the deparser code. If P4\_16 had a direction that meant "input only, but you are allowed to modify your copy locally", we would have used that for headers into the deparsers.

IGM, EGM - inout for the parsers, I believe because we wanted to allow some PSA implementations to initialize the user-defined metadata (some or all of the fields) for the developer, without requiring them to initialize everything themselves in P4 code.